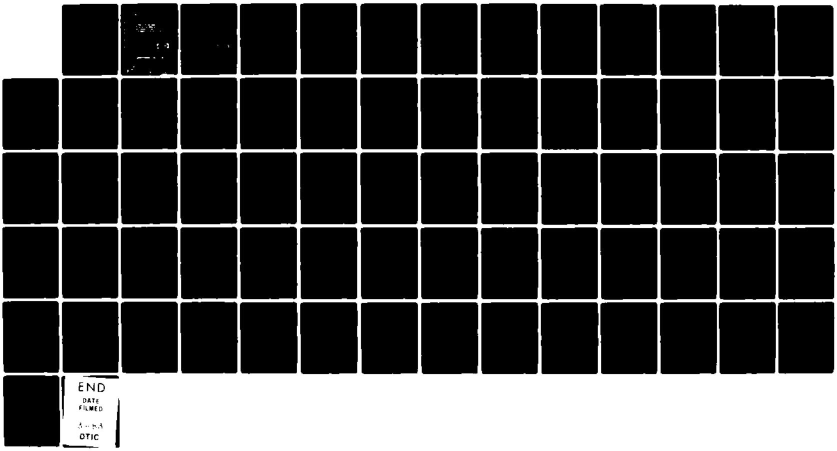
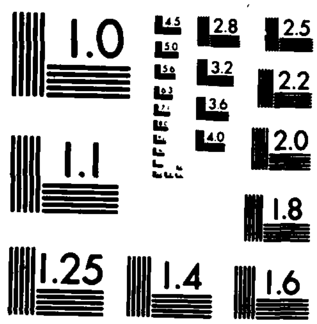
AD-A124 389 ALTERNATIVE IMPLEMENTATION OF AN AUTOCORRELATION UNIT USING TTL-MSI AND N..(U) ILLINOIS UNIV AT URBANA COORDINATED SCIENCE LAB D L HALPERIN 16 OCT 80 R-896 1/1

UNCLASSIFIED N00014-79-C-0424 F/G 9/5 NL





MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

(12)

AD A124389

REPORT [illegible] OCTOBER, 1980 UILU-ENG 80-2228

CSL COORDINATED SCIENCE LABORATORY

ALTERNATIVE IMPLEMENTATION OF AN AUTOCORRELATION UNIT USING TTL-MSI AND NMOS-LSI TECHNOLOGY

[illegible] OF ILLINOIS – [illegible]

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER	2. GOVT ACCESSION NO. AD-A124 387	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) ALTERNATIVE IMPLEMENTATION OF AN AUTOCORRELATION UNIT USING TTL-MSI AND NMOS-LSI TECHNOLOGY		5. TYPE OF REPORT & PERIOD COVERED Technical Panel
		6. PERFORMING ORG. REPORT NUMBER R-896; UILU-ENG 80-2228
7. AUTHOR(s) DANIEL LEE HALPERIN		8. CONTRACT OR GRANT NUMBER(s) N00014-79-C-0424
9. PERFORMING ORGANIZATION NAME AND ADDRESS Coordinated Science Laboratory University of Illinois at Urbana-Champaign Urbana, Illinois 61801		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS
11. CONTROLLING OFFICE NAME AND ADDRESS Joint Services Electronics Program		12. REPORT DATE October 16, 1980
		13. NUMBER OF PAGES 61
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office)		15. SECURITY CLASS. (of this report) UNCLASSIFIED
		15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Approved for public release; distribution unlimited

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

NMOS technology
VLSI design
Autocorrelators

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

As integrated circuit technology progresses towards VLSI, the problem of designing circuits becomes increasingly difficult. In response to these difficulties, design methodologies have been proposed for NMOS LSI with the goal of simplifying the design process. In order to obtain this simplification, very conservative design rules are used. This report, via a case study, examines whether use of these conservative design rules for NMOS LSI results in serious performance degradation compared to an

DD FORM 1 JAN 73 1473

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

20. ABSTRACT (Continued)

equivalent system implemented using SSI/MSI TTL logic.

Several autocorrelator function units are designed using a variety of architectures and technologies. The resulting speed, power, and area are then compared among the designs. Based on this case study, it appears that systems designed in NMOS using a simplified design methodology can indeed be competetive with a system designed using SSI/MSI TTL.

UILU-ENG 80-2228

ALTERNATIVE IMPLEMENTATIONS OF AN AUTOCORRELATION UNIT USING TTL-MSI AND NMOS-LSI TECHNOLOGY

by

Daniel Lee Halperin

This work was supported in part by the Joint Services Electronics Program (U.S. Army, U.S. Navy and U.S. Air Force) under Contract N00014-79-C-0424.

ALTERNATIVE IMPLEMENTATIONS OF AN AUTOCORRELATION UNIT USING TTL-MSI AND NMOS-LSI TECHNOLOGY

BY

DANIEL LEE HALPERIN

B.S., University of Tennessee, 1978

THESIS

Submitted in partial fulfillment of the requirements
for the degree of Master of Science in Electrical Engineering
in the Graduate College of the
University of Illinois at Urbana-Champaign, 1981

Urbana, Illinois

ACKNOWLEDGMENT

I would like to express my sincere appreciation to my advisor, Professor Edward S. Davidson for his invaluable guidance and assistance with my research. I would also like to thank Professor Janak H. Patel for supervising and directing the initial phases of this project and Professor George W. Swenson for originally suggesting the autocorrelator project.

I would most of all, like to thank my fiancee, Beverly Gray, for her support, encouragement, and assistance in finishing this thesis.

TABLE OF CONTENTS

CHAPTER 1

INTRODUCTION

Mead and Conway [1] have set forth a design methodology for designing LSI (Large Scale Integrated) circuits using NMOS (N-channel Metal-Oxide-Semiconductor) technology. The NMOS process is a self-aligning polysilicon gate technology with depletion mode loads. Six masks are required to completely define the circuit on the silicon substrate. These masks can be designed using any of several interactive design packages.

The goal of the Mead and Corway methodology is to allow even inexperienced designers to implement large complex systems. They basically favor a hierarchical approach to the design of systems. The design process begins with the choice of an appropriate algorithm for solving the problem. With the algorithm in mind, the system can be designed. Mead and Conway advocate planning the system as combinations of register-to-register data transfer paths, controlled by finite state

machines. The design at this point can proceed to breaking the system up into cells. By structuring the design in a top down manner, the interaction of cells can be well defined. Therefore, each type of cell can be designed somewhat independently of the rest of the system. The actual circuit level implementation is done following a simple set of design rules. These design rules specify the minimum widths, separations, and overlaps of components patterned on the silicon substrate. Dimensions are always specified in terms of lambda. Lambda is defined to be one half of the minimum feature size. Over the past ten years, the minimum feature size has steadily decreased. There is no reason to assume that this trend will not continue for the next several years. Each fabrication process will have its own value of lambda based on the minimum feature size of the process. As technology improves, minimum feature sizes will decrease and lambda will decrease accordingly. By using dimensionless design rules, the mask design is independent of the particular fabrication process used. All dimensions are strictly in terms of the unit lambda. As minimum feature sizes change, the corresponding change in lambda will automatically scale all linear dimensions to the correct sizes.

When designing cells, Mead and Conway suggest keeping the topology as simple as possible. Combinational logic is implemented using highly regular structures such as PLAs. Finite state machines are implemented as synchronous sequential circuits using a two phase nonoverlapping clock scheme. Information storage is typically done with charge storage on gate inputs (dynamic memory). By following this procedure, simple,

easy to design cells can be created and interconnected to form an entire system. The design rules are straightforward and easy to use. There is no need to use a circuit level simulator to determine the parameters for each transistor. In design methodologies currently practiced in industry, this is not the case. Very complicated design rules are followed and circuit level simulation (which is expensive) is used extensively. Use of Mead and Conway's design rules tend to decrease design time at the expense of design optimization.

1.1 The multiproject chip

The idea behind the multiproject chip (MPC) is to allow several projects to be merged together and implemented as one chip. In this manner it is possible to spread the maskmaking and fabrication costs over several projects. During the fall of 1979, several universities teaching courses in LSI technology took part in MPC79, a multiproject chip experiment in which students designed digital systems in accordance with Mead and Conway's design methodology. The mask information at each school was then encoded into a common language and transfered to a central location over the ARPANET. There, the information was merged and the students projects were fabricated as a multiproject chip. Reference 2 gives a description of the MPC system used and the results of MPC79. In addition, a comparison is done between the costs and turnaround time for designing a system in NMOS as a MPC against designing a comparable system in SSI/MSI TTL (Small Scale Integration/Medium Scale Integration Transistor-Transistor Logic). Most special purpose digital systems in indust y as well as student projects

are built using off-the-shelf TTL parts. The authors of reference 2 show that based on their experience from MPC79, one can indeed design and implement such projects with a turnaround time and cost comparable to designing such a system in SSI/MSI TTL.

1.2 Overview of thesis

It has already been shown that systems can be designed and implemented in NMOS with a turnaround time and cost similar to that for TTL. Nothing however has been said about the performance of the resulting NMOS system. The design methodology and design rules are much simpler than those used in industry. As a result, the design rules have to be very conservative. The simplicity of the design methodology is bought at the expense of performance. A very important question is whether using such a simple design methodology will result in a system with performance vastly inferior to that obtainable from TTL. This thesis examines this question by use of a case study. Several different autocorrelators were designed using both TTL and NMOS logic. These designs results are then compared and conclusions drawn. In particular we will show that NMOS logic designed using Mead and Conway's design methodology can be competitive with TTL logic.

CHAPTER 2

DESIGN OF AUTOCORRELATORS

The need to compute the autocorrelation function of a signal arises in many applications. The high performance autocorrelators designed in this chapter arise from an application in radio astronomy. They allow the real time computation of the autocorrelation function of a high frequency signal derived from radio telescopes. These autocorrelators can also be used for any other application where one needs the autocorrelation function of a high frequency signal. The low cost designs of the autocorrelator perform the same computation but on lower frequency signals.

The sampling theorem tells us that one must sample a signal at twice its highest frequency in order to prevent aliasing. If the signal we are interested in is approximately bandlimited at 25 MHz, then the autocorrelator must be capable of accepting data at a rate greater than or equal to 50 MHz if it is to run in real time. This rate points up

the need for special hardware. Even a large dedicated mainframe computer is incapable of performing the required computations at this rate. In fact, a dedicated mainframe computer is unable to calculate the autocorrelation function except over a small time-lag space at even one MHz. On the contrary, special purpose hardware can be designed and built at a fraction of the cost of a large mainframe computer. As we shall see, this hardware can operate at data rates higher than the 50 MHz mentioned above.

The autocorrelation function $R(j)$ can be considered a measure of how closely a function delayed by j matches itself. The autocorrelation function of $X(t)$ is defined by the following integral:

$$R(j) = \lim_{t \to \infty} \left[\frac{1}{2T} \int_0^T X(t)X(t-j)dt \right] \qquad (1)$$

where $X(t)$ is assumed to be zero for all time t less than zero. We can make the following approximation to $R(j)$ by performing the integration in equation (1) over a finite interval:

$$R(j) \approx \frac{1}{2T} \int_0^T X(t)X(t-j)dt \qquad (2)$$

In our particular situation, we want to design a channel that will compute $R(j)$ for a given time-lag j. Since the system will be digital, the input signal $X(t)$ must be discrete in both magnitude and time and

the integral in equation (2) can be replaced with the following summation:

$$R(j) = \sum_{t=0}^{M} X(t)X(t-j) \qquad (3)$$

Notice that we are ignoring the multiplicative constant in front of the integral in equation (2). This can be viewed as scaling $R(j)$. The magnitude of the input signal $X(t)$ will be quantized to one of four levels and will thus require two bits for representation. Reference 3 discusses some of the issues involved with two bit quantization in correlators. In particular, it shows that for signals with a low degree of correlation, two bit quantization only degrades the signal-to-noise ratio of the correlator to 88% of the value for a continuous correlator.

The two bit representation to be used consist of a sign bit and a magnitude bit (sign-magnitude form). The quantization encoding scheme is defined in Table 1. We want to calculate $R(j)$ for values of j from 0 to some number P. The time-lag space spanned by the autocorrelator is 0 to P and we therefore need P+1 channels. Channel j accepts as inputs $X(t)$ and $X(t-j)$ and over some time period M, computes $R(j)$. At the end of the time period M, each channel shifts out its $R(j)$ value, a 23 bit result. The channels are then cleared and begin to compute the next value of $R(j)$. The 23 bit result is required so these designs will be compatible with an existing design. A separate clock is used for the channel shift register and the rest of the system so that the rate at which $R(j)$ is shifted out of the channel can be independent of the

Table 1. Input encoding

	INPUT VOLTAGE RANGE			
	vin<=-V	-V<vin<=0	0<vin<V	V<=vin
SIGN BIT	1	1	0	0
MAGNITUDE BIT	1	0	0	1
WEIGHTING FACTOR	-2	-1	1	2

channel's data rate.

The block diagram in Fig. 1 illustrates the channel configuration. Two parallel P-bit shift registers are used to provide X(t-j). Some type of data acquisition system such as a mini-computer collects the values of R(j) and either stores them on some medium such as tape or performs further computations on them. The data acquisition system will have to be fast enough to service all P+1 channels before the next values of R(j) are calculated to prevent losing data. Notice that the present value of R(j) will be repetitively shifted out of the channel while the next value of R(j) is being calculated.

The system under investigation for effective design is the channel. External to the channel will be additional hardware such as the shift register and the control section. We are not concerned with this external circuitry. Instead our focus is on the channel itself. Five different channel designs are examined and contrasted. Two of these designs consist of commercially available TTL SSI/MSI components. They are built entirely of Schottky and low power Schottky parts. The Schottky technologies are generally considered the most advanced of the TTL logic families. The other three designs are NMOS integrated circuits. The NMOS designs use a two-phase clocking scheme. In order to decrease the delay times in the high performance NMOS designs, the clock voltages PHI 1 and PHI 2 use zero volt and seven volt logic levels. The shift register clocks, THETA 1 and THETA 2, use zero volts and five volts. For the low cost NMOS designs, all clocks use zero volt

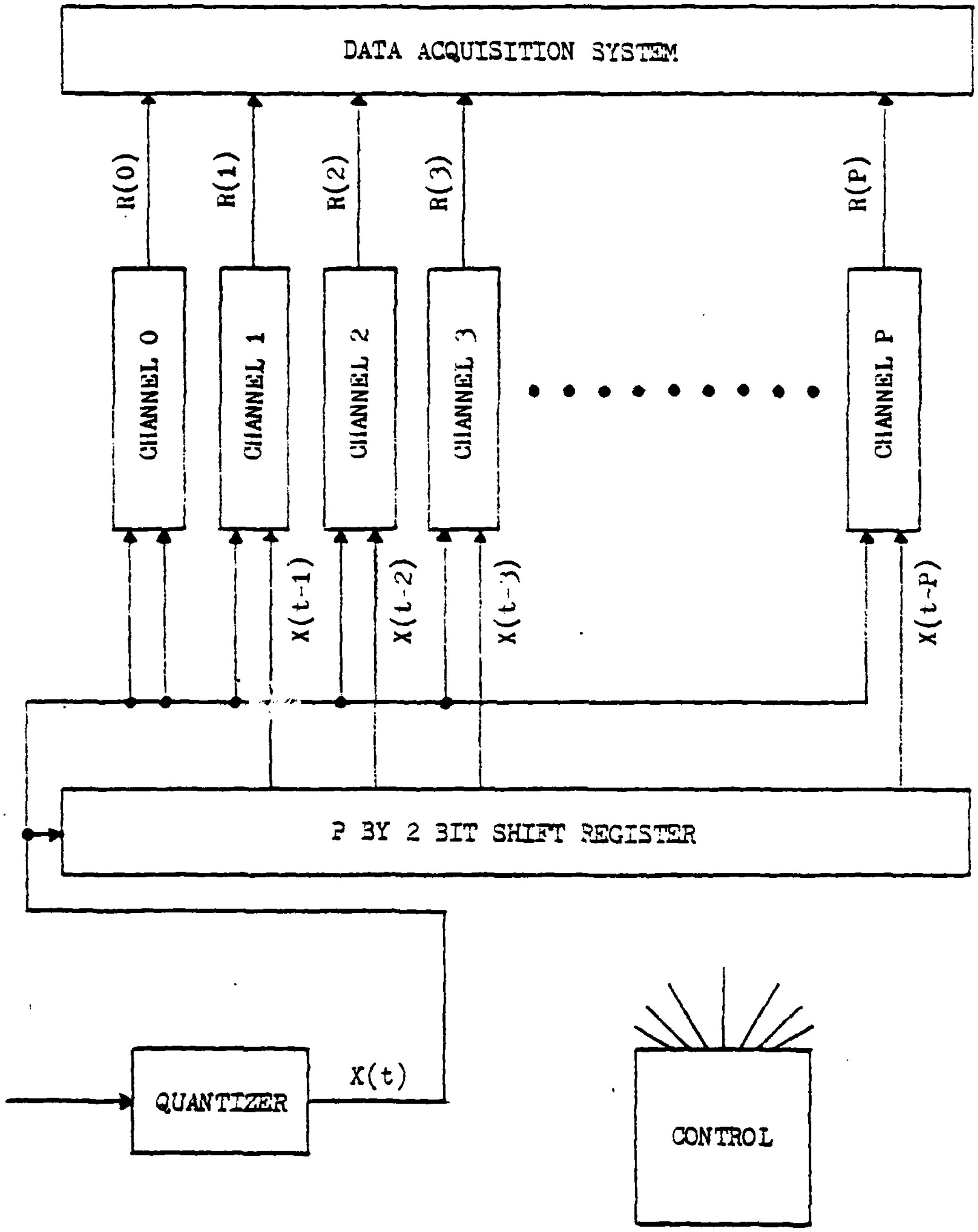


Fig. 1 Autocorrelator channel configuration

and five volt levels. Note that VDD equals five volts. The five designs can be split into two different groups. In the first group the primary consideration is to minimize power consumption and chip count (chip area for the NMOS design). In the second group, the goal is to maximize the data rate. All systems in the second group should be able to operate at a data rate of 50 MHz or greater.

2.1 Low cost designs

The low cost designs attempt to minimize the power consumption and chip count (chip area for the NMOS design), even at the expense of speed. These designs represent an extreme and as we will later see in section 3.3, a rather modest increase in hardware can result in a dramatic improvement in performance.

2.1.1 Low cost TTL design

Fig. 2 shows a gate level diagram of the low cost TTL design. This design is a very straightforward implementation of equation (3). The data X(t) and X(t-j) are clocked into D flip-flops. The outputs of the D flip-flops corresponding to the sign bits are exclusive-ORed together to determine if the product of X(t) and X(t-j) is positive or negative. The output of this exclusive-OR gate is logic one to indicate a negative result and logic zero to indicate a positive result. If the two magnitude bits are both logic one, then the magnitude of the product is 4. This condition is detected by the AND gate. If one magnitude bit is logic one and the other is logic zero then the magnitude of the product is 2. This condition is detected by the exclusive-OR gate. Likewise,

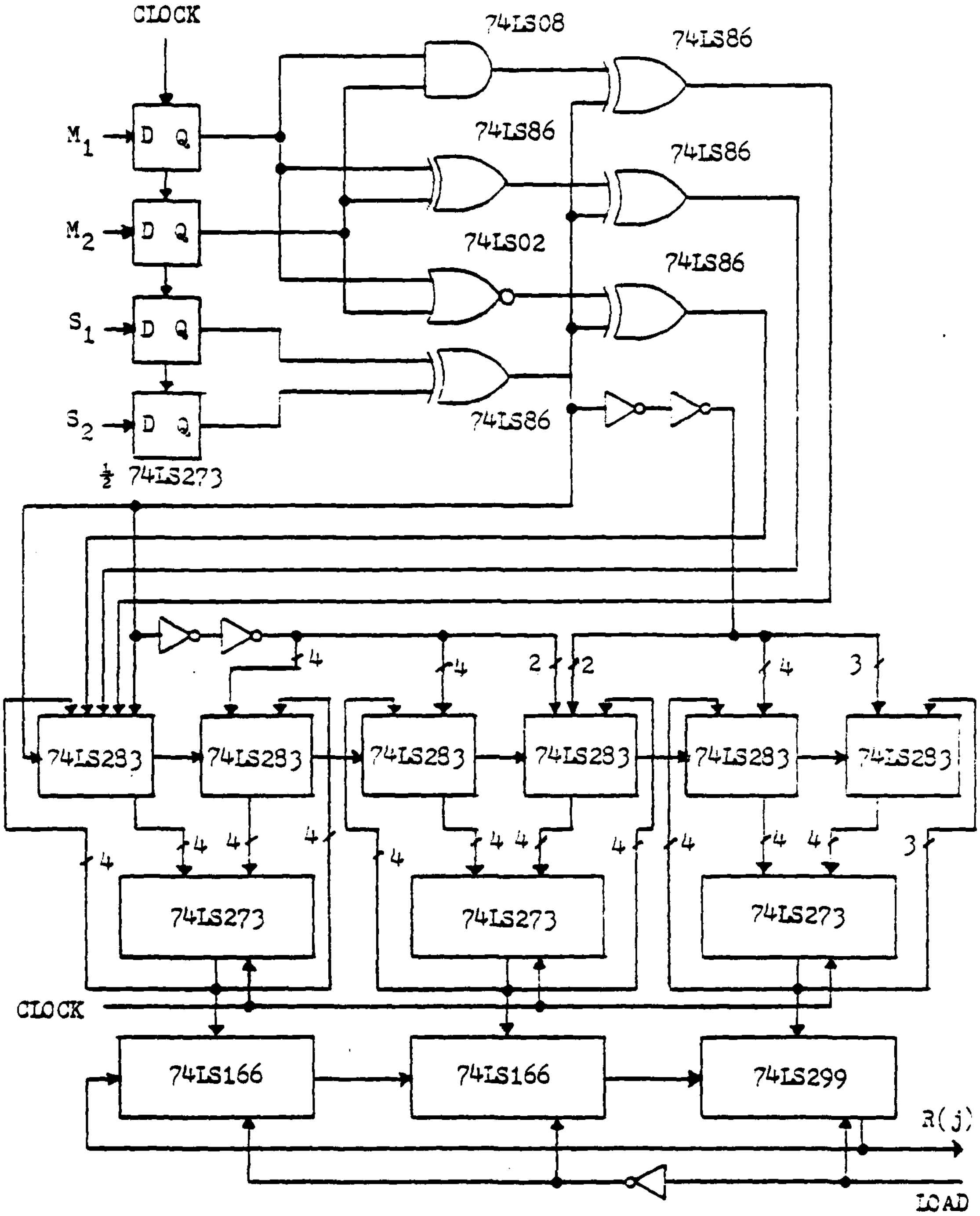


Fig. 2 Low cost TTL design

if both magnitude bits are logic zero, then the magnitude of the product is 1. This condition is detected by the NOR gate.

The outputs of these three gates form a three bit binary word which equals the absolute value of the product. The product is converted to a 23 bit ones' complement number by exclusive-ORing the three bit value with the sign of the product. This forms the three least significant bits of the 23 bit value. The 20 most significant bits are the same as the sign bit. The six 74LS283 chips form a 24 bit adder although the most significant bit of the adder (bit farthest to the right in Fig. 2) is not used since we are only interested in a 23 bit result. The sign bit is also input to the carry in of the least significant 74LS283 adder chip so that the addition will be in two's complement arithmetic.

Three 74LS273 chips are used as an accumulator register which accumulates the summation of equation (3). Notice that the most significant bit of the accumulator is also not used. The inputs to the adder are the accumulator register output and the 23 bit version of the product. These two inputs are summed together and the result is stored back in the accumulator.

The two 74LS166 chips (parallel-load, serial out) and 74LS299 chip (parallel-load, parallel out) are connected to operate as a 23 bit circular shift register for output. The output register is loaded with the contents of the accumulator when the channel has finished calculating R(j). After the shift register is loaded, the accumulator register is cleared and computation begins on the next value of R(j).

The shift register will continue to shift out R(j) until it is loaded with a new value.

2.1.2 Low cost NMOS design

In most respects the low cost NMOS design is very similar to the low cost TTL design except that the logic is implemented with PLAs. Fig. 3 shows the block diagram of the NMOS implementation. The two sign bits are latched into the EXOR PLA cell. This cell exclusive-ORs the sign bits to find the sign of the product. The cell labeled MADD serves two different functions. First of all it forms the product of X(t) and X(t-j). Next it sums this product with the three least significant bits of the accumulated sum. These three bits are stored internally in MADD since the structure of a PLA allows outputs to be stored efficiently within the array. MADD also computes a carry out that is cascaded into the next cell which is labeled ADDT.

ADDT is a three bit adder which in a manner similar to MADD, also stores its three bit result internally. Therefore, the MADD cell and the seven ADDT cells together form a 24 bit adder with a built-in accumulator. The MADD cell adds the three low order bits of the product to the three low order bits of the accumulated sum. The ADDT cells add the 20 high order bits of the product to the 20 high order bits of the accumulated sum. The 20 high order bits of the product are always the same as the sign bit. Therefore the 20 high order bits of the product are equal and one input to the ADDT cell is all that is required to specify the product. As was the case with the TTL design, the most

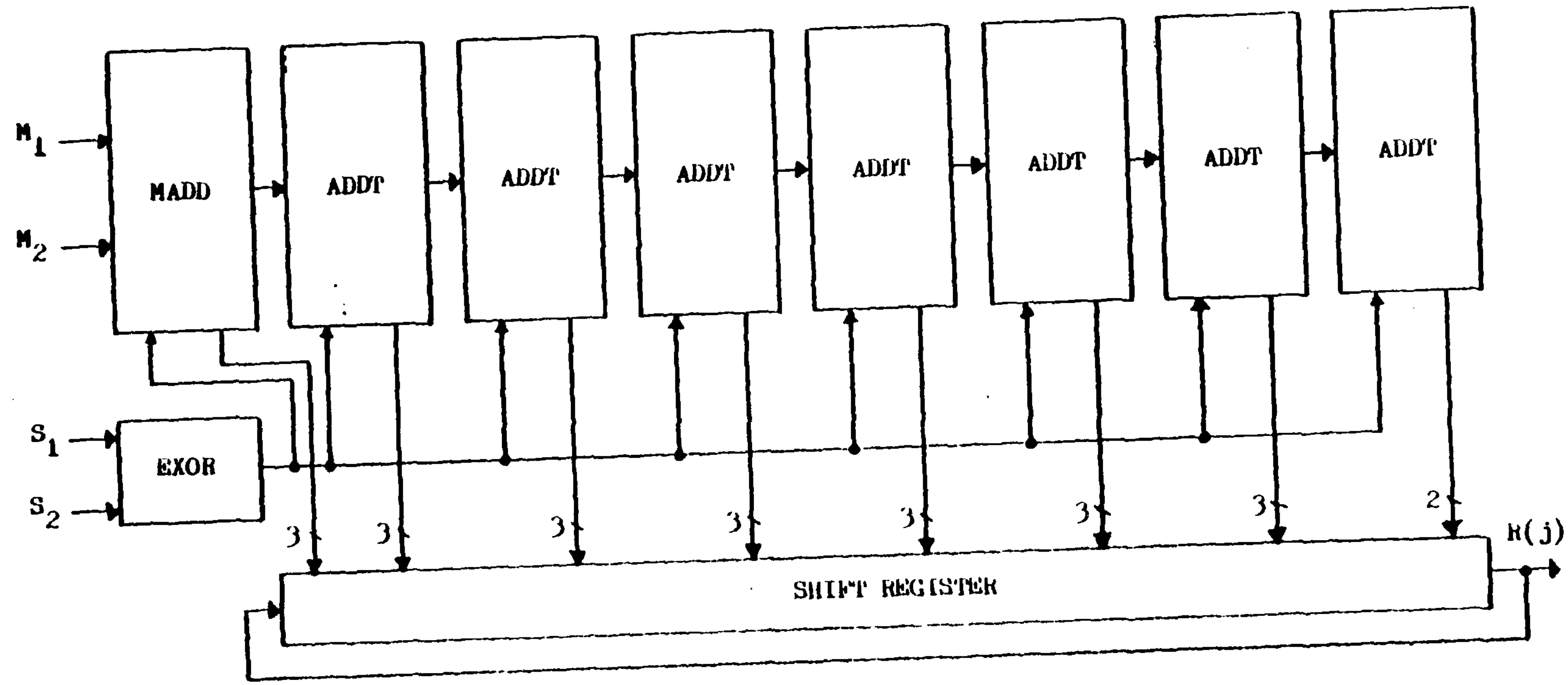


Fig. 3 Low cost NMOS design

significant bit of the adder is unused. When the channel has completed calculating $R(j)$, the result is loaded into the circular shift register and shifted out repetitively until the next value of $R(j)$ is calculated.

2.2 High performance designs

In order for the high performance designs to operate at the maximum possible speed, it is necessary to use an architecture that will allow parallel operations. The most obvious architecture to accomplish this is one that exploits pipelining. In addition, it is desirable to replace the operation of addition with a simpler and hence faster operation such as counting.

The product of $X(t)$ and $X(t-j)$ can only take on one of six possible values; -4, -2, -1, 1, 2, and 4. Therefore we will use six counters. Each counter will correspond to one of the six possible values of the product. For each product value $X(t)X(t-j)$, the counter corresponding to that value will be incremented. After all samples have been received and counted, the summation is evaluated. The sum is formed by multiplying each of the counts by their associated scale factors (-4, -2, -1, 1, 2, or 4) and summing them together.

Fig.4 shows a block diagram of the high performance design. The two magnitude bits are presented to the magnitude multiplier while the two sign bits are exclusive-ORed together to calculate the sign. One of the three output lines of the magnitude multiplier will be activated to indicate the magnitude of product. These three output lines go to a three to six demultiplexer. The demultiplexer is controlled by the

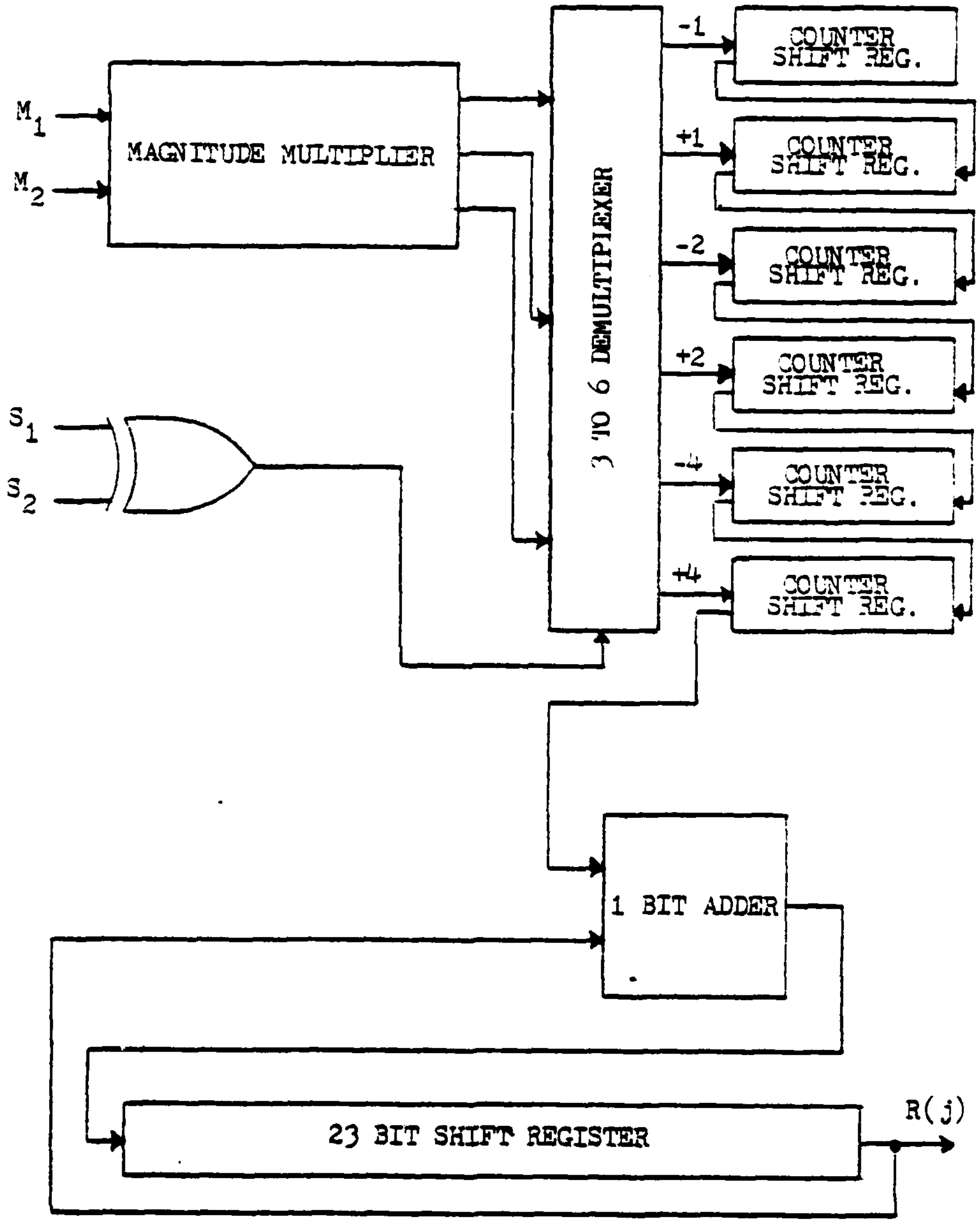


Fig. 4 Block diagram of the high performance design

output of the exclusive-OR logic. One and only one of the six output lines of the demultiplexer will be activated at any one time. The outputs of the demultiplexer drive the inputs of the six counters. Each counter contains a shift register to shift out the count as well as logic to multiply the count by the correct scale factor. Since the scale factors are all powers of two, this logic is quite simple. The six shift registers are concatenated to allow serial addition of all six counts. A one bit adder is used to sum the six counts serially and a 23 bit shift register is provided to accumulate the sum. This register also serves as the channel output register.

2.2.1 High performance TTL design

Fig. 5 shows the gate level design for the magnitude multiplier, the exclusive-OR logic for determining the sign of the product, and the demultiplexer. Note that this logic is constructed with three pipeline stages. For maximum throughput, each stage consists of only one level of NAND gates. NAND gates were chosen because they have a smaller propagation delay than NOR gates in TTL technology. The pipeline registers are made up of D flip-flops using 74S74 chips. They were chosen for their short set-up times and propagation delays. The logic shown for the magnitude multiplier is a NAND implementation of the following boolean expressions:

$$ONE = (M1 + M2)'$$

$$TWO = M1 \oplus M2$$

$$FOUR = M1 \cdot M2.$$

Notice the exclusive-OR operation on the sign bits is also accomplished

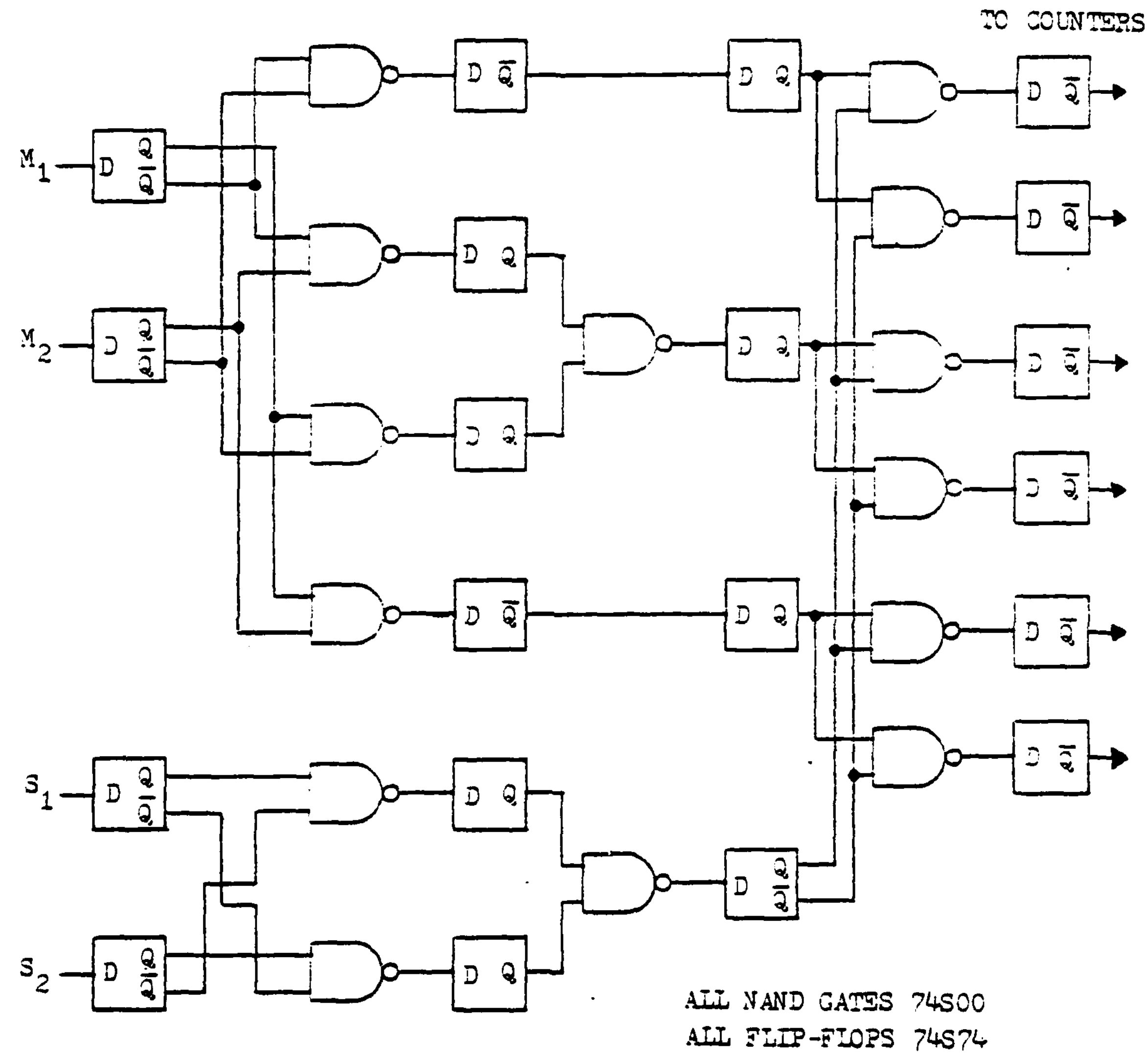


Fig. 5 Input section for high performance TTL design

with a NAND realization. The demultiplexer function is performed in the third pipeline stage. The output of this logic is input to the counters.

Fig. 6 shows the design of one of the six identical counters. The five 74X197 chips are used to perform the count operation. These are asynchronous four bit ripple counters. The least significant counter chip is a 74S197 chip since the first stage of the counter has to respond to a very high frequency. Each stage however acts to divide this frequency by 16 so succeeding stages do not need to be as fast. Therefore the remaining four chips are 74LS197's to reduce power consumption.

A 20 bit shift register is formed from the two 74LS165 chips and the 74LS194A chip. Low power Schottky chips are used because the data is shifted through the shift registers relatively slowly. The exact rate depends on the data input rate of the system. The channel can only accept 524,287 inputs ($2^{19} - 1$) without risk of overflowing the counters. Therefore, the time it takes to sum the six counts together and shift the result out must be less than 524,287 divided by the data input rate. We will probably want to run the shift registers at a somewhat higher speed than this to allow the data acquisition system to collect the R(j)'s serially instead of all in parallel.

The three input NAND gate serves two purposes. If the same input is applied repeatedly to the channel, the output from the demultiplexer is a level signal. However, the counter is edge-triggered. Thus if

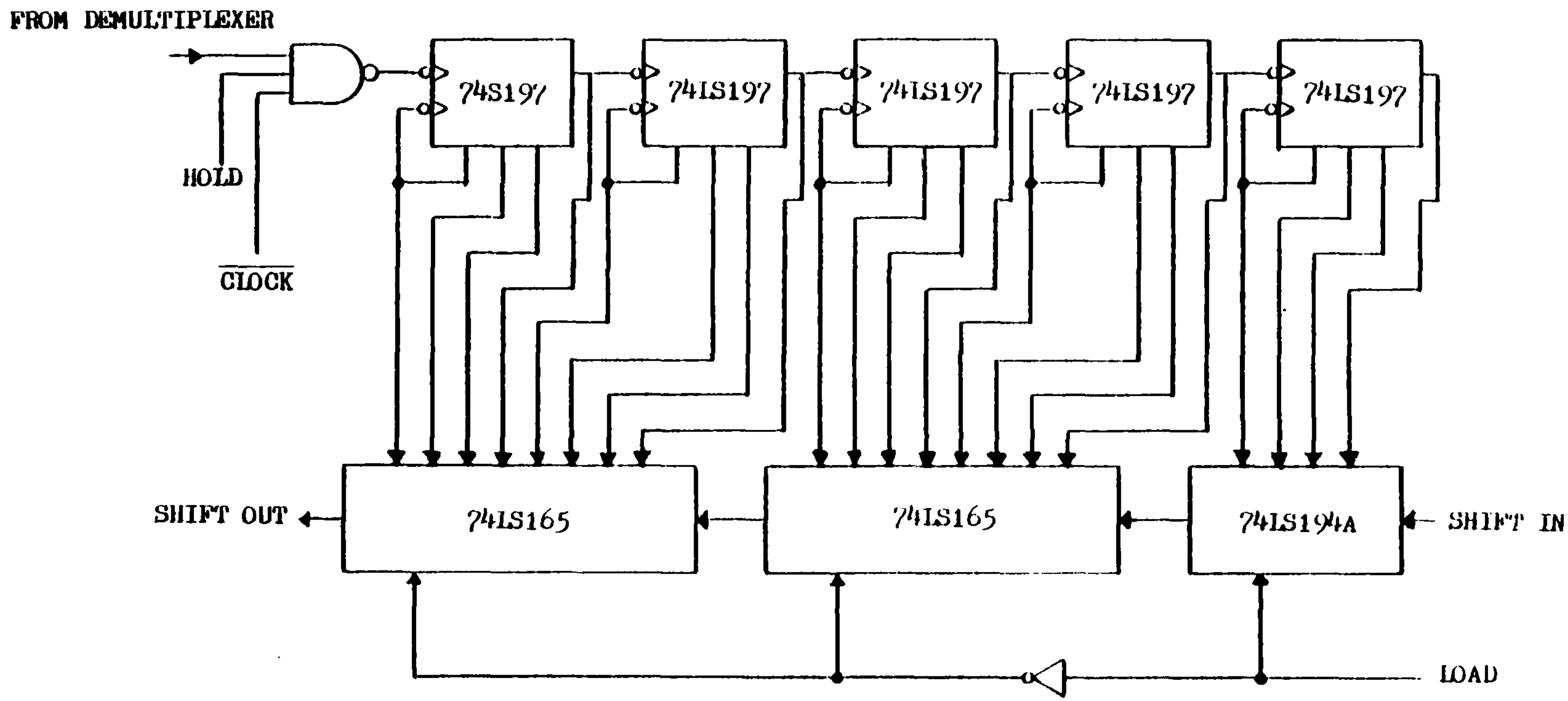


Fig. 6 Counter for high performance TTL design

only the demultiplexer output were input to the counter, the counter would count the product once instead of each time it occurs since it would only see one input transition. To prevent this from happening, the data from the demultiplexer is NANDed with CLOCK′ to convert the level signal into a pulsed signal. The other purpose of the NAND gate is to easily disable the counter. After the 524,287 inputs have been applied to the channel it is necessary to give the counters a sufficient time for all carries to ripple the entire length of the counter without further inputs. When the HOLD signal goes to logic zero, the output of the NAND gate is forced high and the counter is disabled since it is negative edge triggered.

Fig. 7 shows the logic for correcting the sign of each count and multiplying it appropriately. The shift register associated with each 20 bit counter has three bits appended onto it, making the shift registers 23 bits instead of 20 bits. These additional bits are appended to be in the most significant position on the 1′s counters. On the 2′s counters, two bits are appended to the most significant position and one bit is appended to the least significant position. The 4′s counters have one bit appended to the most significant position and two bits appended to the least significant position. The two bits appended to the 4′s counters effectively multiply the values in the 4′s counters by 4. Likewise, the bit appended to the least significant position of the 2′s counters effectively multiply the values in the 2′s counters by 2. The six 23 bit shift registers are cascaded together so that the least significant bit of one register is shifted through an inverter to

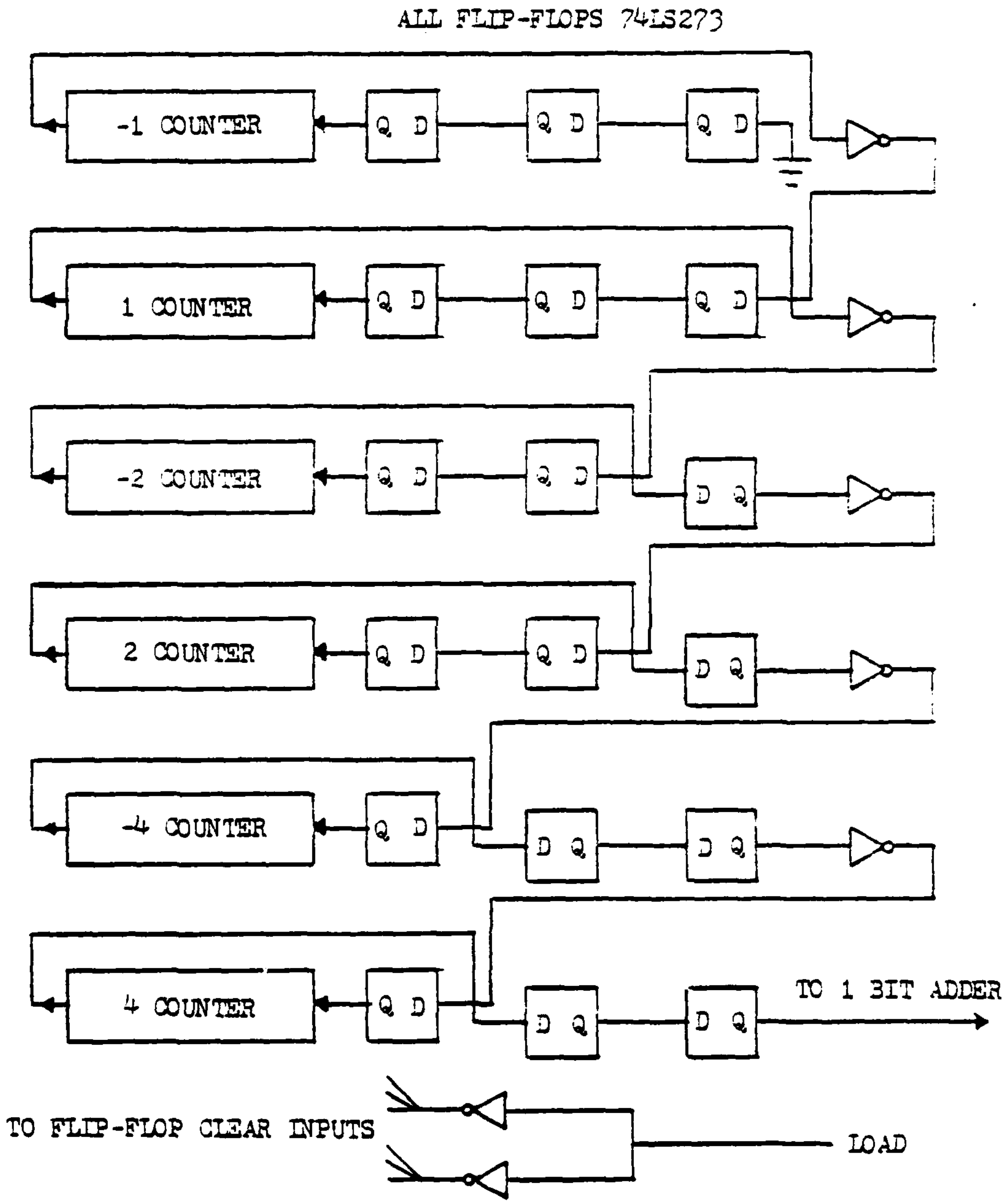


Fig. 7 Sign correction and scaling for high performance TTL design

become the most significant bit of the next register. The six shift registers now form one 138 bit shift register. The 138 bit shift register is shifted into the one bit serial adder shown in Fig. 8. An inspection of the 138 bit shift register shows that as the count values snake their way through, those count values that are supposed to be positive pass through an even number of inverters while those values which are supposed to be negative pass through an odd number of inverters. Therefore, as the contents of the 138 bit shift register are shifted into the adder, they are ones' complement values corresponding to the count values multiplied by the correct scale factor and of the correct sign. The 23 bit shift register associated with the adder accumulates the final sum. Notice that the serial adder scheme correctly implements the end-around carry required by ones' complement arithmetic.

2.2.2 High performance synchronous NMOS design

The synchronous NMOS design is very similar to the TTL high performance design. The major difference is in the counters. Unlike the counters used in the TTL design, the NMOS counters are synchronous rather than asynchronous. In addition, the logic for the magnitude multiplier although quite similar is implemented using NMOS dynamic storage registers as the pipeline registers and NOR gates instead of NAND gates. NOR gates in general have a smaller propagation delay than NAND gates in NMOS technology.

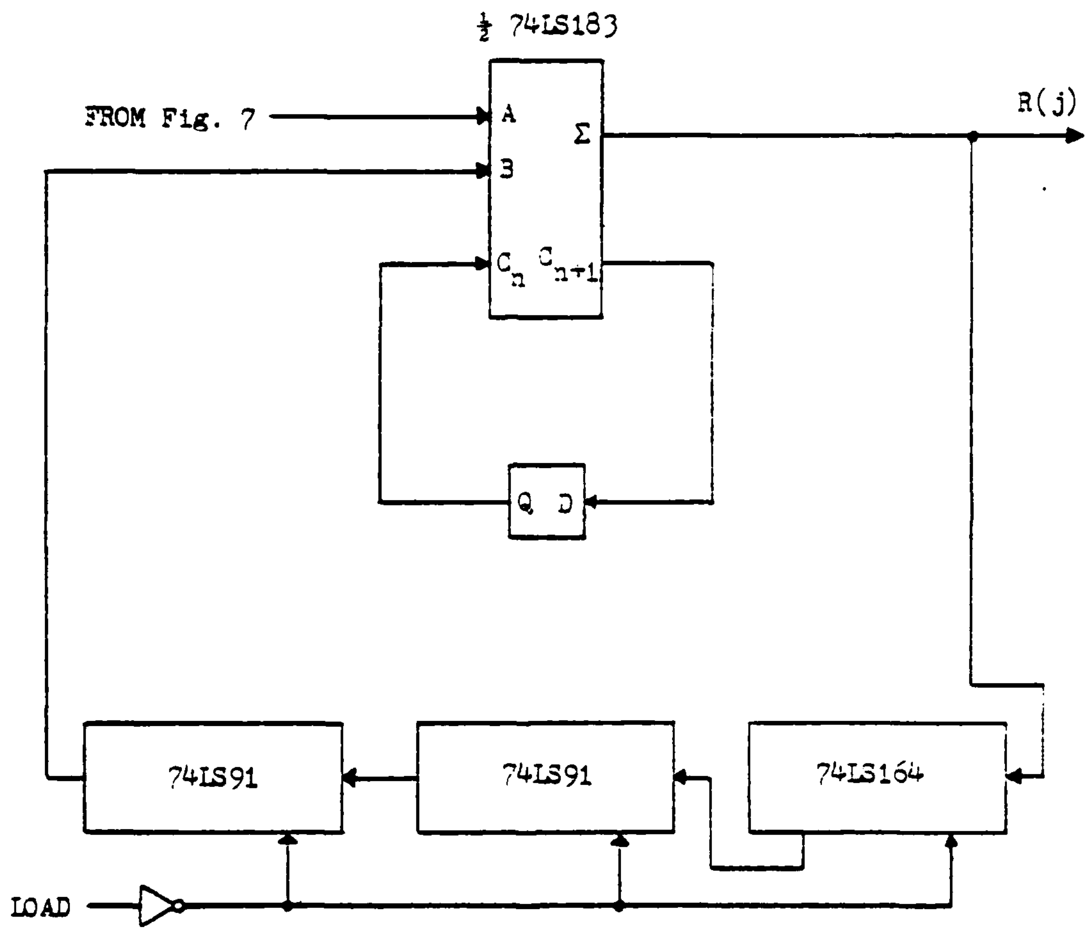


Fig. 8 Final accumulation of R(j) for high performance TTL design

A block level diagram of the NMOS channel layout is shown in Fig. 9. It consists of an input section, an array of counters, and a summation section. Fig. 10 shows a gate level diagram of the INOR cell which corresponds to the magnitude multiplier, the sign exclusive-OR gate and the demultiplexer. Fig. 11 shows the gate level diagram for the HOLD cell. This cell accomplishes one of the purposes of the NAND gate at the input of the TTL counters. Since the NMOS counters are synchronous, it is not necessary for the inputs to be pulsed. Therefore, one of the purposes of the NAND gate in the TTL design is unnecessary in this design. It is, however, still necessary to be able to disable the counter when we have finished counting and are waiting for carries to propagate. The HOLD cell serves this purpose. Whenever the HOLD line is a logic one, the output of the HOLD cell will be a logic one and the counter is disabled. Since it is important for the data passing through the HOLD cell to remain in sync with the rest of the system, the HOLD cell is pipelined. Fig. 12 shows the gate level diagram of the COUNT cell. It is a pipelined version of a ripple carry counter. Whenever Cin′ is a logic zero, the counter counts. Notice that when a carry is generated, it is not presented to the next stage of the counter until the next clock period. Contained in the COUNT cell is a dynamic shift register. The remaining logic is functionally identical to the logic used in the TTL design. The shift registers are padded to 23 bits in the same manner. The serial adder is implemented as a PLA.

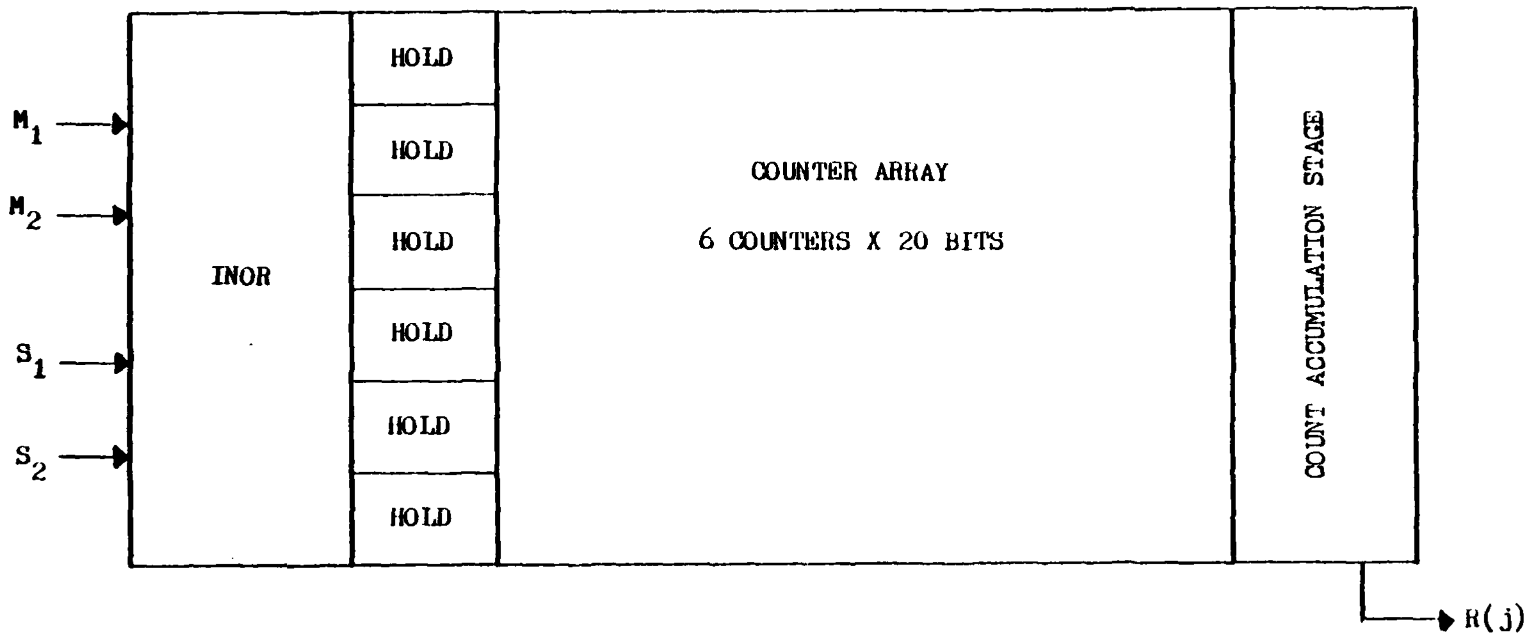


Fig. 9 Block diagram of high performance NMOS design

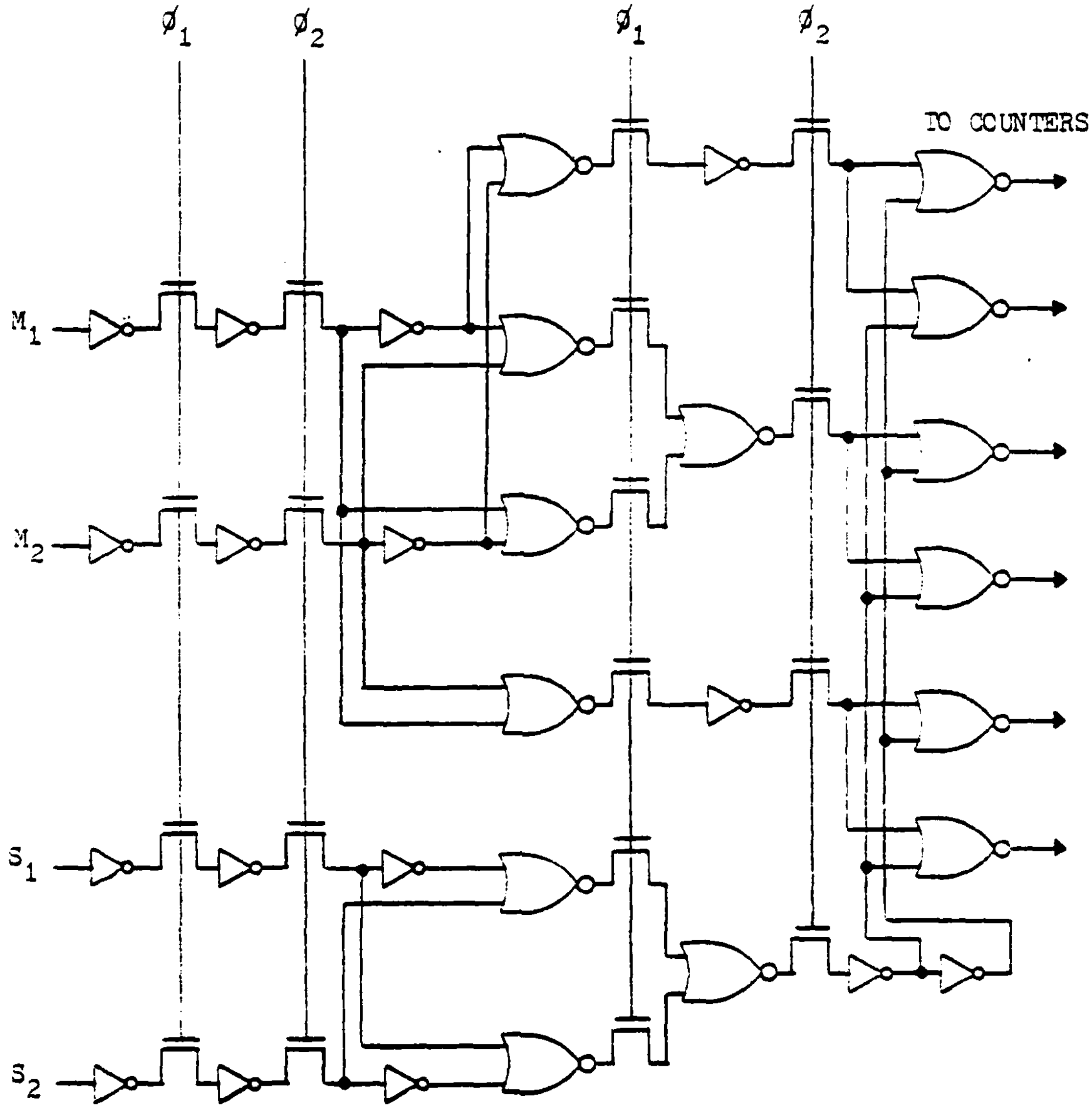


Fig. 10 Input section for high performance NMOS design

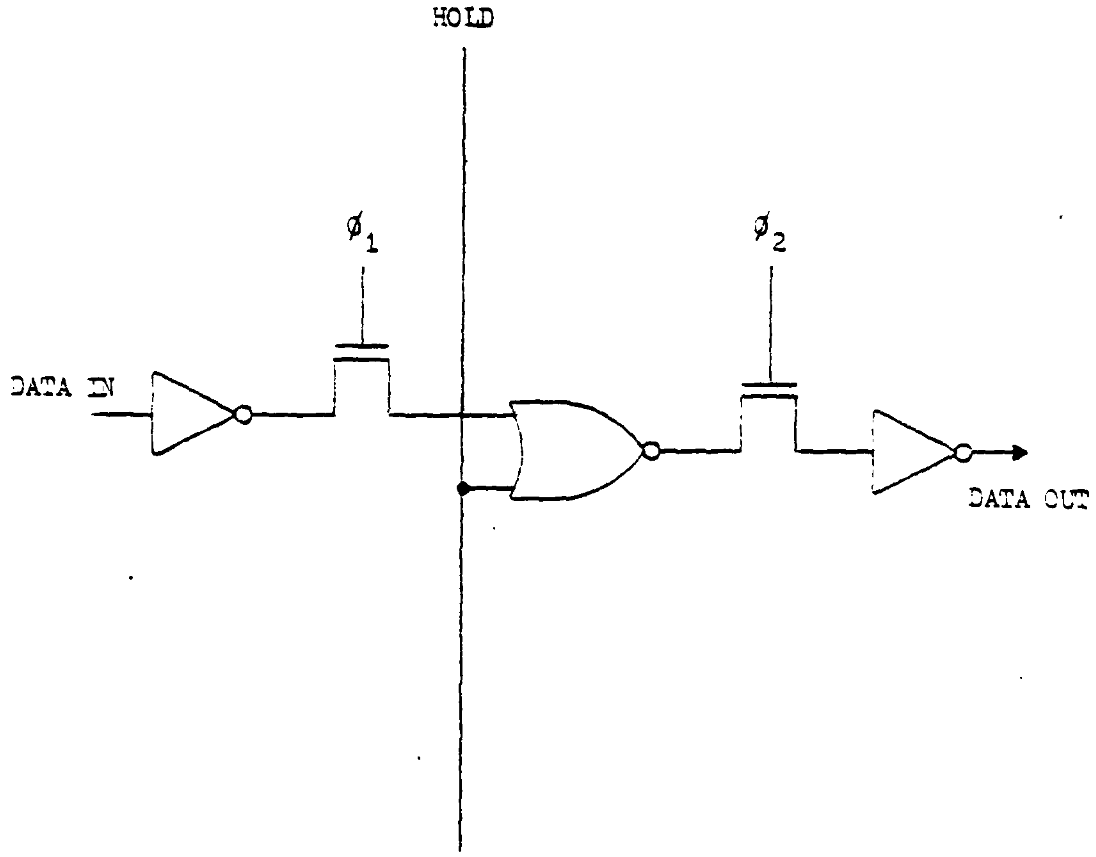


Fig. 11 Hold cell for high perf. synchronous NMOS design

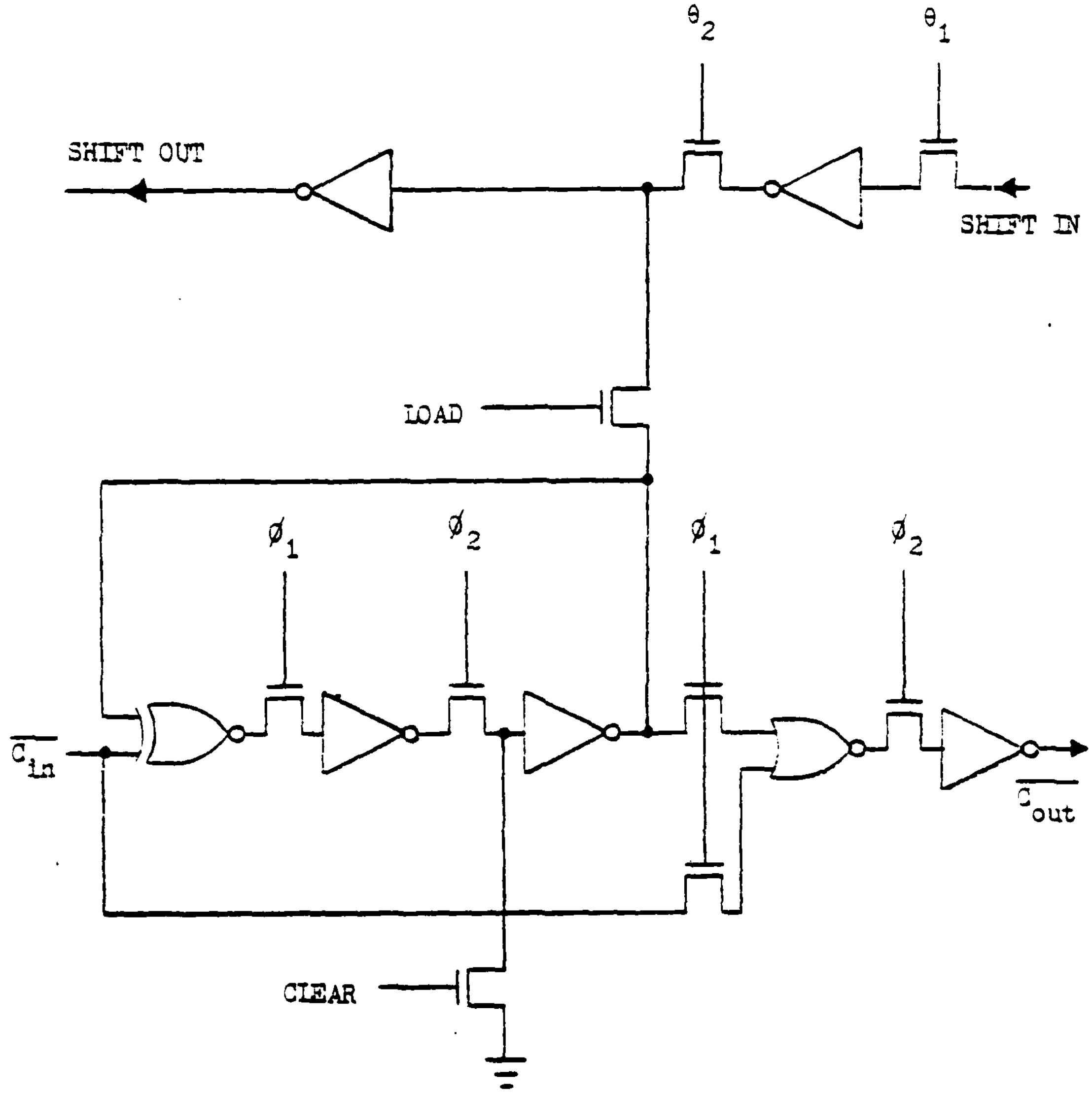


Fig. 12 Count cell for high perf. synchronous NMOS design

2.2.3 High performance asynchronous NMOS design

The asynchronous design is identical to the synchronous design with the exception of the implementation of the counters. Since the counters are asynchronous, the NMOS design methodology discussed in chapter 1 is violated. In the next chapter we examine the costs as well as the benefits of abandoning this design methodology.

The counters in this design are implemented as fundamental mode T flip-flops. The standard techniques of designing fundamental mode circuits are used. For reasons of economy, the combinational logic of the T flip-flop is built using two NMOS four to one multiplexers; one for each feedback loop. Fig. 13 shows a gate level diagram of the asynchronous counter cell. The shift register is basically the same as the shift register in the synchronous counter cell. Just as was the case with the TTL design, the lowest order stages of the counter have to handle high frequency inputs while the higher order stages have lower frequency inputs. The cell in Fig. 13 is able to respond to high frequency inputs but consumes fairly large amounts of power. Fig. 14 shows another counter cell design which is functionally identical to the counter of Fig. 13, but uses different output drivers which are much slower and consume much less power. The first counter cell is used as the first four stages of the counter where its ability to handle high frequencies is needed. The second counter cell is used for the remaining 16 stages. Since the counters in this design are asynchronous, it is necessary for the inputs to be pulsed just as in the TTL design. The cell called HPUL shown in Fig. 15 serves this purpose.

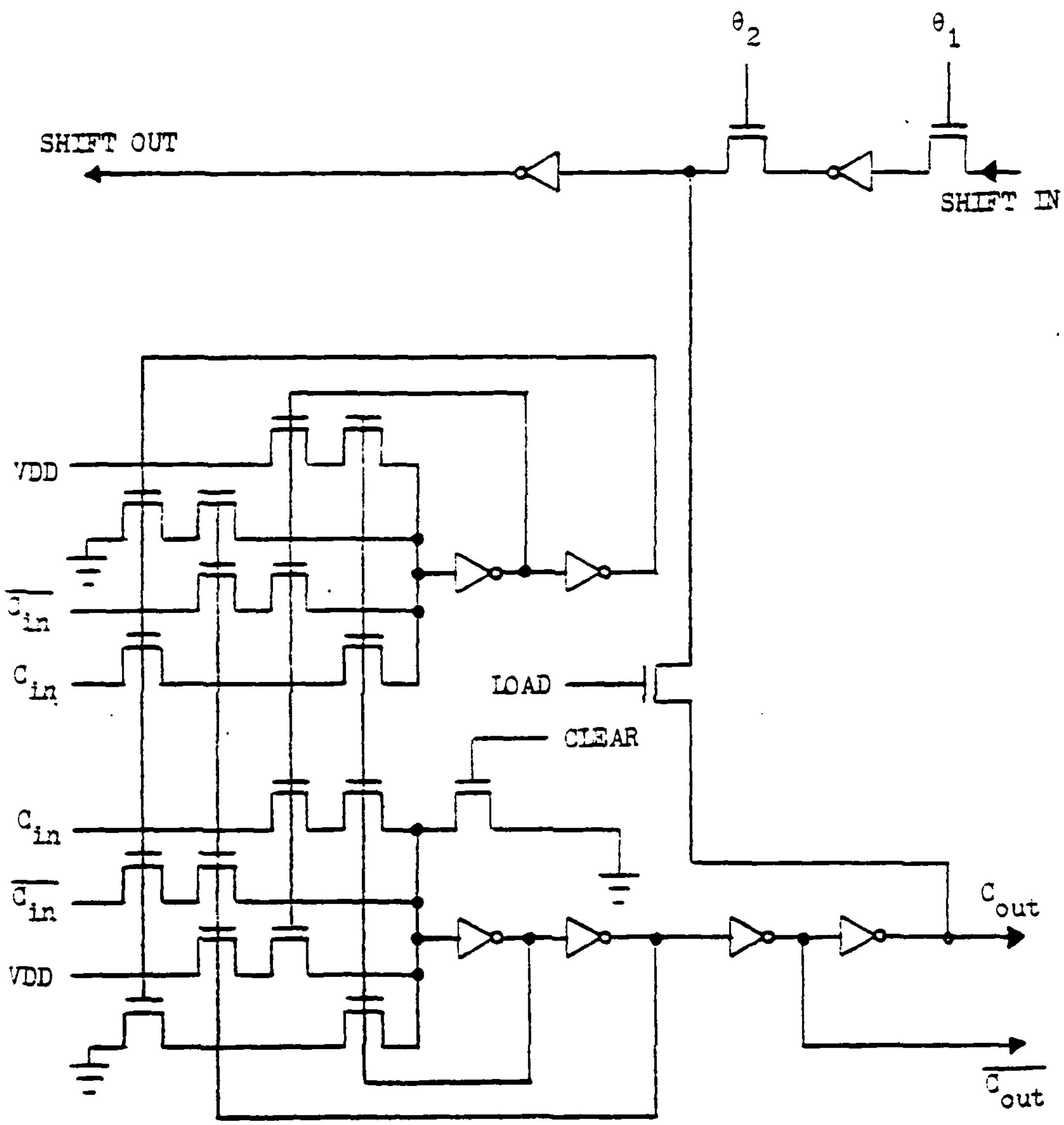


Fig. 13 High speed asynchronous NMOS counter cell

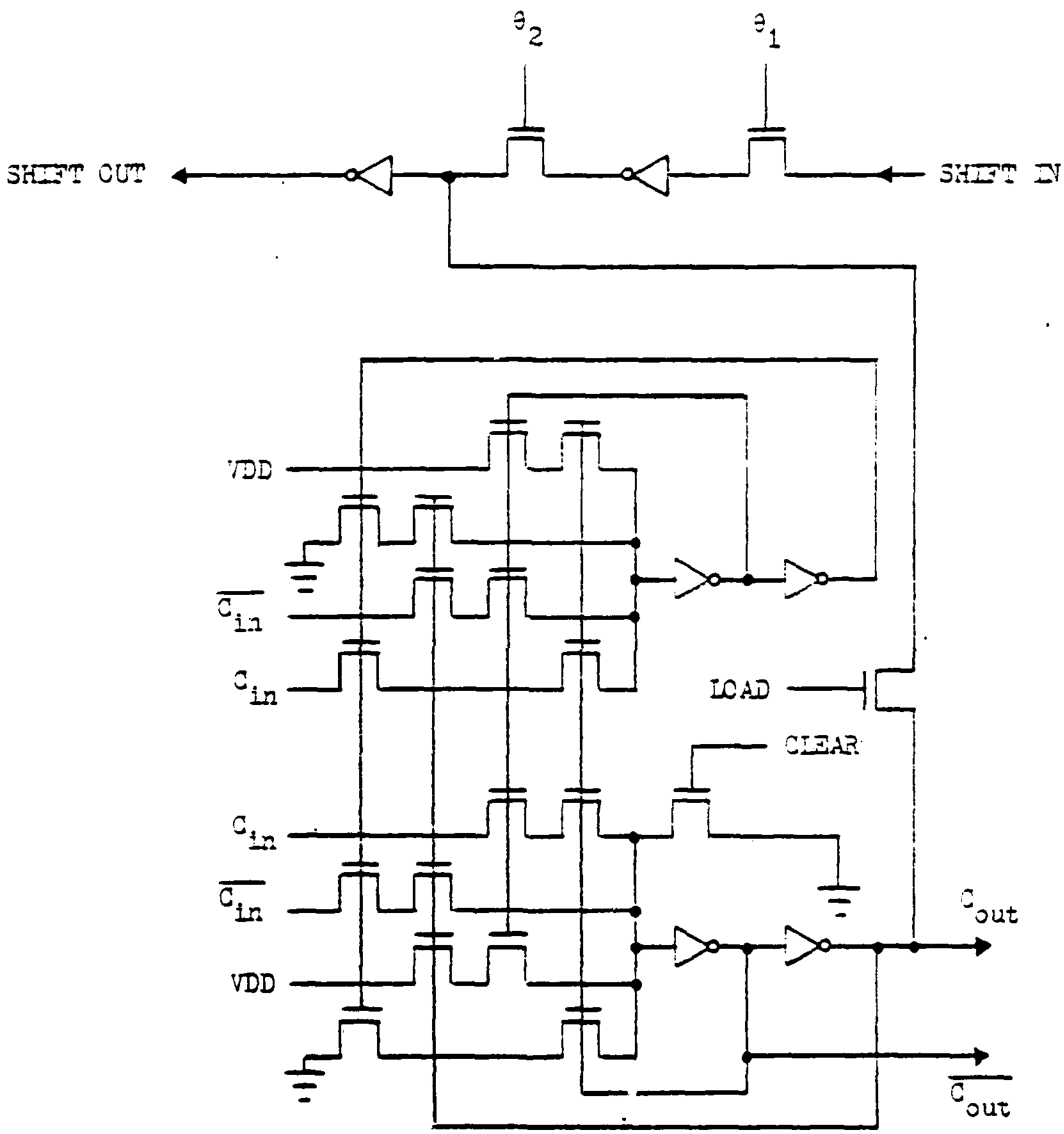


Fig. 14 Low power asynchronous NMOS counter cell

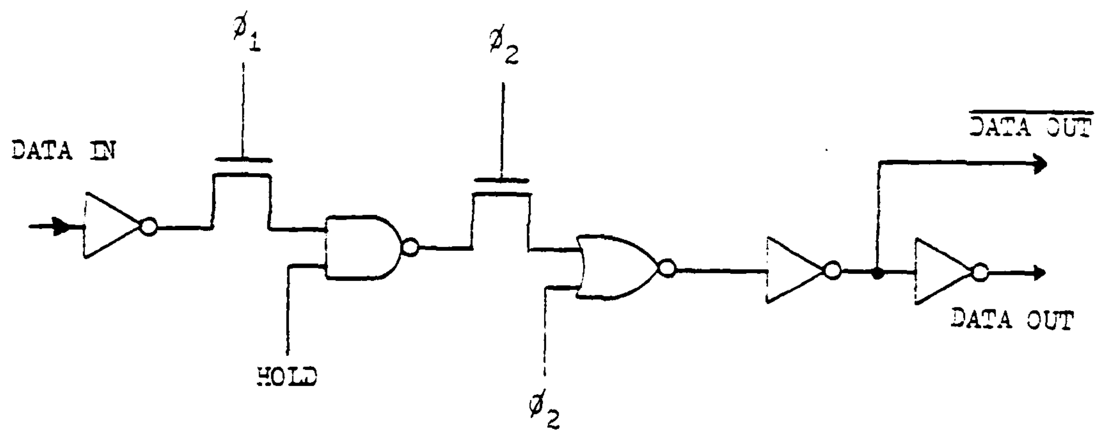


Fig. 15 HPUL cell for high perf. asynchronous NMOS design

Furthermore it disables the counter to allow carries to propagate.

CHAPTER 3

COMPARISON OF DESIGNS

There are many possible parameters that can be used to compare the architecture and implementation of one system to another. The three we use in this chapter are power consumption, data rate, and area. The first two parameters are used to directly compare TTL SSI/MSI designs with NMOS designs. This comparison is not really meaningful for the third parameter, area. When we speak of the area of a TTL design, we mean the number of chips used to implement the design. Even though not all TTL chips used are the same size, this is still a reasonable measure of design size. On the other hand, when we speak of the area of a NMOS design, we are referring to the actual silicon area used by the design. Since the NMOS designs are all implemented as one chip systems, it is a foregone conclusion that any TTL SSI/MSI design will necessarily occupy more space than a NMOS design. Therefore, physical area comparisons are only done between designs using the same type of logic.

It is also desirable to be able to examine the tradeoffs between these parameters in each design. One way of contrasting the tradeoffs between data rate and power consumption is to use the power-delay product (also known as the speed-power product). It is simply the product of the power consumption and delay time between successive data inputs (reciprocal of the data rate). This measure is quite commonly used to compare technologies at the gate level but as each of the different system designs are performing the same computation, it should be just as useful for system level comparisons as for gate level comparisons. The power-delay product is a valid measure of the power consumption-data delay tradeoff if these two parameters are inversely related, that is their product will be constant as they are varied. This inverse relationship is not always true for NMOS. If however, an entire system is implemented with NMOS gates using minimum size transistors, it is possible to reduce the system's delay roughly by a factor of two by doubling the power consumption. This is assuming of course that the resulting power dissipation of the chip, and current density of any of the conductors does not become unacceptable. In addition, we are assuming that the fanout of all gates is relatively small. The area of the new gate can approximately equal the area of the old gate. This can be done over a range of at least 4 to 1. A graph on page 5-3 of reference 4, implies that this inverse relationship is also true for TTL logic.

For example, suppose we have a module which consumes one watt of power, occupies one square meter, and can accept data at a rate of one Hz. If we need a system that has twice the data rate, we have argued that we can do this using roughly the same area, but it will require twice the power. By using this new module, we will consume two watts of power, occupy one square meter, and operate at a two Hz data rate. The power-delay product will be one joule as it was in the original module.

However, it is often possible to use identical modules operating in parallel to increase the rate at which a computation is performed. If the overhead required to coordinate the modules is small, and there is sufficient parallelism in the computation, then we can expect that the use of n modules would increase the data rate of the system by approximately n. Assuming that the same computation can be done with a negligible amount of overhead in parallel, then we can also achieve the two Hz data rate by operating two of the original modules in parallel. Then the data rate will be two Hz, the power consumption will be two watts, but now the system will occupy too square meters. The power-delay product is once again one joule but the second design requires twice the area of the first design.

If area is an important consideration, then the power-delay product alone is a poor figure of merit. Perhaps the simplest (although by no means the only) figure of merit is power-delay-area product. Since this product involves the area, it is only valid for comparisons of TTL to TTL or NMOS to NMOS designs.

Accordingly we shall refer to power, delay, area, power-delay product, and power-delay-area product when comparing designs.

3.1 How evaluation data was obtained

The performance of the NMOS designs was primarily determined through simulation of cells using version 2F.1 of SPICE [5]. Some cells are too large to simulate economically. The performance of these cells was determined from estimates based on the simulation of parts of the cell in question. Lambda, one half the minimum feature size, was taken to be 2.5 microns. The parameters for the transistor models are as follows:

ENHANCEMENT	DEPLETION
VTO = 1.12	VTO = -5.2
KP = 3.12E-5	KP = 3.46E-5
GAMMA = .266	GAMMA = .914
CGSO = 5E-12	CGSO = 5E-12
CGDO = 5E-12	CGDO = 5E-12
TOX = 1E-7	TOX = 1E-7
NSUB = 2.5E14	NSUB = 3E15
NSS = -22E10	NSS = 80E10
XJ = 5E-7	XJ = 5E-7
TPG = 1	TPG = 1
UO = 910	UO = 800
UCRIT = 8E3	UCRIT = 8E3.

In addition, the following parameters were used to derive values for circuit capacitances and resistances:

POLYSILICON CONTACT RESISTANCE = 2500 OHMS

DIFFUSION CONTACT RESISTANCE = 1400 OHMS

BUTTING CONTACT RESISTANCE = 4000 OHMS

POLYSILICON CAPACITANCE = 1E-16 FARADS/SQUARE MICRON

DIFFUSION CAPACITANCE = 4E-16 FARADS/SQUARE MICRON

METAL CAPACITANCE = 3E-17 FARADS/SQUARE MICRON.

These parameters with the exception of KP, were given in a communication to participants in the spring 1980 multiproject chip. The value of KP was changed so the channel resistance of an on MOSFET transistor would match the value of 10,000 OHMS/SQUARE given in reference 1. The values were obtained by Mike Beaver of Hewlett-Packard. The parameters were either measurements or estimates for the HP NMOS process used for the 1979 multiproject chip set. As was stated in the communication, "The following is a suggested list of parameters for the BDM MOSFET model, which may be reasonable for typical transistors designed at lambda = 2.5 microns." The communication goes on to state, "In any case, they (the parameters) are not guaranteed to any level of accuracy for MPC580 (spring 1980 multiproject chip), as there is no history data for the process. Don't assume this data will be accurate for MPC580 - we simply do not yet know!" This statement tends to indicate that the parameters are at best approximate and we should be careful not to assume that they accurately predict the performance of future fabrication runs. Furthermore even if there were no doubts concerning the parameters, there is still the question of how accurately the models used by the SPICE program can simulate the actual performance of an integrated

circuit.

The parameter values shown above were used with the SPICE program because it was felt they were the best method available to estimate performance data on integrated circuits. It is hoped the performance results are representative of what can typically be achieved with current technology rather than predictive of the performance of any given fabrication run. The results should be interpreted with these facts in mind. Although I have no reason to doubt the results, only time will tell how accurate they are.

The TTL data was obtained from reference 4. Because of the nature of the NMOS data, it was felt that typical data for the TTL chips instead of worst case data should be used for comparison. In a few cases, typical data was not given for some parameter of a TTL chip. In those situations, worst case data was used. Since the autocorrelator consists of a large number of channels, it is assumed that unused gates on the chips of one channel can be shared with adjacent channels. For this reason, the chip counts contain fractions of a chip.

3.2 Design comparison results

Table 2 shows the performance data for all five designs. Several trends are immediately obvious. All of the NMOS designs have a much lower power-delay product than any of the TTL designs. The advantage that TTL offers is higher speed. The high performance TTL design is capable of running at a data rate of over 83 MHz while the fastest NMOS design is limited to a data rate of just over 71 MHz. This modest

Table 2. Evaluation of basic designs

TTL DESIGNS	(1) CHIP COUNT	(2) POWER WATTS	(3) DELAY SECS	(2)X(3) JOULES	(1)X(2)X(3) J-CHIPS
SCALE FACTOR	X1	X1	$X10^{-9}$	$X10^{-9}$	$X10^{-6}$
LOW COST	15.2	1.35	150.	203.	3.09
HIGH PERFORMANCE	71.9	8.21	12.0	98.5	7.08

NMOS DESIGNS	(1) AREA $METER^2$	(2) POWER WATTS	(3) DELAY SECS	(2)X(3) JOULES	(1)X(2)X(3) $J\text{-}METER^2$
SCALE FACTOR	$X10^{-6}$	X1	$X10^{-9}$	$X10^{-9}$	$X10^{-15}$
LOW COST	2.75	.183	116.	21.2	58.2
SYNC. HIGH PERF.	10.1	2.36	14.0	33.0	334.
ASYNC. HIGH PERF.	10.6	.204	20.0	4.08	43.2

increase in speed however is bought at a very high price in terms of power. The power consumption of the high performance TTL design is nearly three and one half times that of the equivalent NMOS design. Overall, the synchronous high performance NMOS design has a power-delay product which is roughly one third of the equivalent TTL design.

If a lower data rate can be tolerated, the asynchronous NMOS design looks very promising. Its power-delay product is the lowest of any design. Although this design can only run at 50 MHz, it has a power-delay product which is 4% of the high performance TTL design and 12% that of the synchronous NMOS design. It appears that designs using asynchronous logic can under some circumstances offer substantial benefits over designs using synchronous logic. This advantage however, will not always exist and seldom will the benefits be as great as they are in this example.

In particular, very complicated sequential machines are often quite difficult to design as asynchronous circuits. The combinational logic for asynchronous circuits is usually more complex than that for synchronous circuits. In general, however, the combinational logic of simple cells such as the T flip-flop, requiring one or two feedback paths can be implemented with NMOS multiplexers. These multiplexers simplify design topology and have a much lower power-delay product than a corresponding synchronous cell. The use of multiplexers will result in slower cells with much lower power consumption. Higher speed can be obtained by using standard gates but these gates employ more active

pullups and thus consume much more power and area. In general, the power-delay product and particularly the power-delay-area product will be higher for a design using standard gates then a design using multiplexers.

Asynchronous designs are most advantageous in situations where different parts of a system are required to operate at vastly different speeds. It should be pointed out, however, that the logic design of an asynchronous cell will be much more difficult than that for a synchronous cell. With an asynchronous cell it is necessary to worry about races and hazards which are not a problem with synchronous cells. If essential hazards appear to be a potential problem as they usually are in fundamental mode circuits, iterative simulation and addition of delay to particular feedback paths may be necessary to achieve correct operation.

Based on power-delay product, the low cost TTL design is significantly worse than the high performance TTL design. If area is included by using the power-delay-area product, then the low cost TTL design is significantly better than the high performance design. In NMOS the low cost design has a slightly better power-delay product than the synchronous design and a dramatically better power-delay-area product than the synchronous design. As mentioned previously, the asynchronous design has the best power-delay product. It also has the best power-delay-area product although the low cost design comes close.

In most cases, the cost of a design will be very closely tied to the area and power parameters. Therefore, it would seem reasonable that the low cost designs would be used in cost sensitive applications while the high performance designs would be used in those applications where high performance is important enough to justify the additional cost.

3.3 Toward a cost effective design

The designs up to this point are oriented toward extreme objectives. The high performance designs are meant to operate at the highest possible data rates while the low performance designs are designed to have the lowest possible power consumption and area. With the exception of the high performance asynchronous design, no attempt has been made to tradeoff these factors against each other. Traditionally, the most cost effective designs tend to be those in which such tradeoffs have been made. In particular, if we build a system and attempt to maximize speed at the expense of all other factors, we would tend to find that the incremental cost to obtain a given increase in speed grows rapidly as the speed of the system increases. Therefore we expect the most cost effective design to have an area, power consumption, and data rate that are intermediate values rather than extremes. The rest of this chapter is concerned with more cost effective autocorrelator designs. The metrics we will use for cost effectiveness are power-delay product and power-delay-area product.

One crude method of improving a design might be to implement it in a faster (or lower cost) technology without modifying the design. This procedure was carried out for both the low cost and high performance TTL designs. The low cost TTL design was reimplemented with regular Schottky chips instead of low power Schottky chips. This substitution is done everywhere except the shift registers as the shift registers have no bearing on the data rate of the system. The high performance design was reimplemented with low power Schottky chips replacing the regular Schottky chips.

Another approach is to modify the architecture of one of the designs. The low cost designs in both TTL and NMOS are prime candidates for such a modification. In both designs, the total delay time is dominated by the time required for the carries to ripple from one stage of the adder to the next. By pipelining the adder, the delay time can be cut dramatically.

For the low cost TTL implementation, all that is necessary to accomplish this redesign is ten latches to hold the carries and sign between the stages shown in Fig. 2 and four more latches to hold the data inputs to the first stage. After all the data is received, we need to give carries a chance to propagate through the pipeline. To do this we must force the four data inputs of the lowest stage to zero. We can accomplish this with four AND gates.

In the NMOS design of Fig. 3, this pipelining is very easy to achieve since the logic is implemented using PLAs. It is trivial to pipeline the output of any PLA into the input of any other PLA. This scheme works very nicely for the carries. The sign bit from the exclusive-OR PLA however must be pipelined using latches. It is also necessary to use latches to delay M1 and M2 one clock period before they are presented to MADD. The only real difficulty is in allowing the carries to propagate through the pipeline when we are finished counting. Fig. 16 shows the required control logic to allow final carry propagation. This logic deactivates the PHI 2 clock signal before it gets to the MADD PLA and thus keeps the output of this cell from changing. In addition, the logic forces the carryout of MADD and the sign bit from the exclusive-OR PLA low. The power dissipated by this extra circuitry is quite modest for both the TTL and NMOS designs.

Table 3 shows the results for all nine designs considered. The low cost TTL design using the regular Schottky logic and the high performance TTL design using the low power Schottky logic have the worst power-delay product of any of the designs. The power-delay-area product of the low cost design with regular Schottky logic, however, is much better. On the other hand, the high performance design with low power Schottky has a very poor power-delay-area product.

The modified low cost TTL design has the best power-delay product and power-delay-area product of all the TTL designs. The modified low cost NMOS design has the second best power-delay product, only slightly

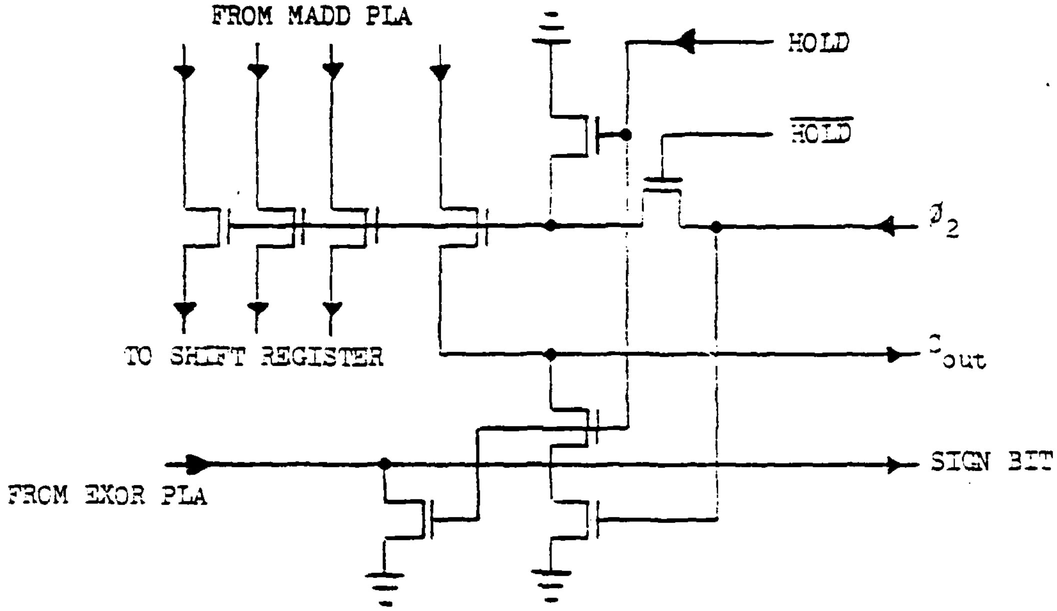


Fig. 16 Logic to allow carries to propagate in modified low cost NMOS design

Table 3. Final design evaluation

TTL DESIGNS	(1) CHIP COUNT	(2) POWER WATTS	(3) DELAY SECS	(2)X(3) JOULES	(1)X(2)X(3) J-CHIPS
SCALE FACTOR	X1	X1	$X10^{-9}$	$X10^{-9}$	$X10^{-6}$
LOW COST	15.2	1.35	150.	203.	3.09
HIGH PERFORMANCE	71.9	8.21	12.0	98.5	7.08
LOW COST WITH HIGH SPEED PARTS	15.2	3.73	98.0	366.	5.57
HIGH PERFORMANCE WITH LOW POWER PARTS	71.9	5.11	59.5	304.	21.9
ENHANCED LOW COST	16.8	1.54	58.0	89.3	1.50

NMOS DESIGNS	(1) AREA $METER^2$	(2) POWER WATTS	(3) DELAY SECS	(2)X(3) JOULES	(1)X(2)X(3) $J\text{-}METER^2$
SCALE FACTOR	$X10^{-6}$	X1	$X10^{-9}$	$X10^{-9}$	$X10^{-15}$
LOW COST	2.75	.183	116.	21.2	58.2
SYNC. HIGH PERF.	10.1	2.36	14.0	33.0	334.
ASYNC. HIGH PERF.	10.6	.204	20.0	4.08	43.2
ENHANCED LOW COST	2.87	.187	22.0	4.11	11.8

larger than the asynchronous design's value. Its power-delay-area product however is only 27% that of the asynchronous design.

The results of this example seem to show that choice of technology and architecture are not separable and should be considered together in the course of designing a system.

CHAPTER 4

CONCLUSIONS

The autocorrelator case study demonstrates quite clearly that the simplified design methodology proposed by Mead and Conway can result in NMOS systems with very respectable performance compared to TTL systems. The only measure examined in which TTL has an advantage over NMOS is in speed. This advantage however is not great and would most likely only be important in those systems required to operate at the highest possible speeds, even at a premium in cost.

On the average, the power consumption of TTL designs were five or six times greater then the power consumption of comparable NMOS designs. The power-delay product also emphasizes the advantages of NMOS. The power-delay product of the worst NMOS design is still 37% of the power-delay product for the best TTL design. The power-delay product for the best NMOS design is 5% of the value for the best TTL design.

In addition, several other conclusions can be drawn from this case study concerning architecture. One of the most dramatic results was that by ignoring the synchronous logic constraint of our design methodology, it may be possible to build systems with much lower power-delay products. We also saw that compromise designs, that is designs that attempt to tradeoff parameters can have much lower power-delay products and power-delay-area products then designs that attempt to maximize just one parameter.

There are many factors not considered in this case study that might be very important for a particular application. For instance, the two phase clock required by the NMOS designs might be a disadvantage if the clock had to be generated off the chip. Another big disadvantage might be the fact that the two phase clock for the high performance NMOS designs had to operate at a voltage above VDD. On the other hand, distribution of the control and clocking signals to a NMOS system does not present a resistive load to the signal drivers as a TTL system would.

In the future, we can look forward to improved performance of NMOS circuits. As minimum feature sizes are reduced, lambda will decrease accordingly. If we assume supply voltages are kept constant, as lambda is decreased by a factor of r, delay time will decrease by the same factor r and area will decrease by r squared. Power consumption, however, will remain constant. Therefore, if lambda is scaled down by a factor of r, the power-delay product will also decrease by a factor of

r. Power-delay-area product will decrease by a factor of r to the third power.

If in addition supply voltage is scaled down by a factor of r, then delay is still decreased by a factor of r, area is still decreased by a factor of r squared, but now power is also decreased by a factor of r squared. This means that for a system with a scaled supply voltage, its power-delay product will decrease by a factor of r to the third power and power-delay-area product will decrease by a factor of r to the fifth power.

MPC79 was implemented with a lambda of 2.5 microns. In MPC580, some of the designs were implemented with a lambda of 2 microns. The power supply voltage in both cases were five volts. The NMOS designs of this thesis were all evaluated using a lambda of 2.5 microns. If the evaluation was repeated using a lambda of 2 microns, we could expect the power-delay product of the NMOS designs to be only 80% of the values we obtained for lambda = 2.5 microns. The high performance synchronous design would then be capable of operating at a data rate greater than the high performance TTL design.

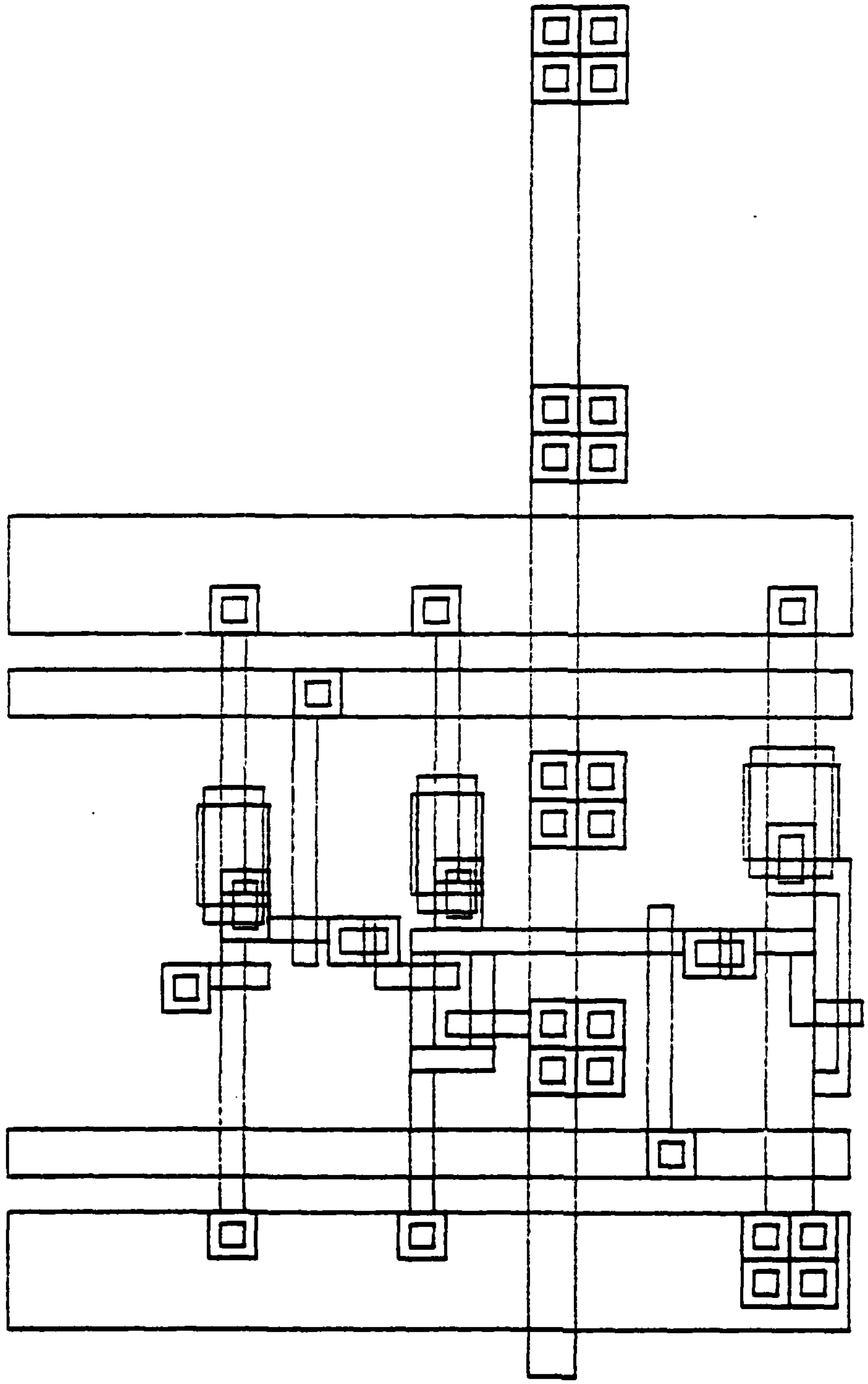
We can expect to see even greater performance increases in NMOS circuits of the future. It is highly unlikely that SSI/MSI TTL will be improved at a similar rate. Bipolar technologies do not scale in the same manner as MOS technologies. In any event, SSI/MSI TTL systems are limited in performance by delays due to driving signals off the chip. Future improvement in such TTL systems is expected to be focussed on

decreases in power consumption rather than speed.

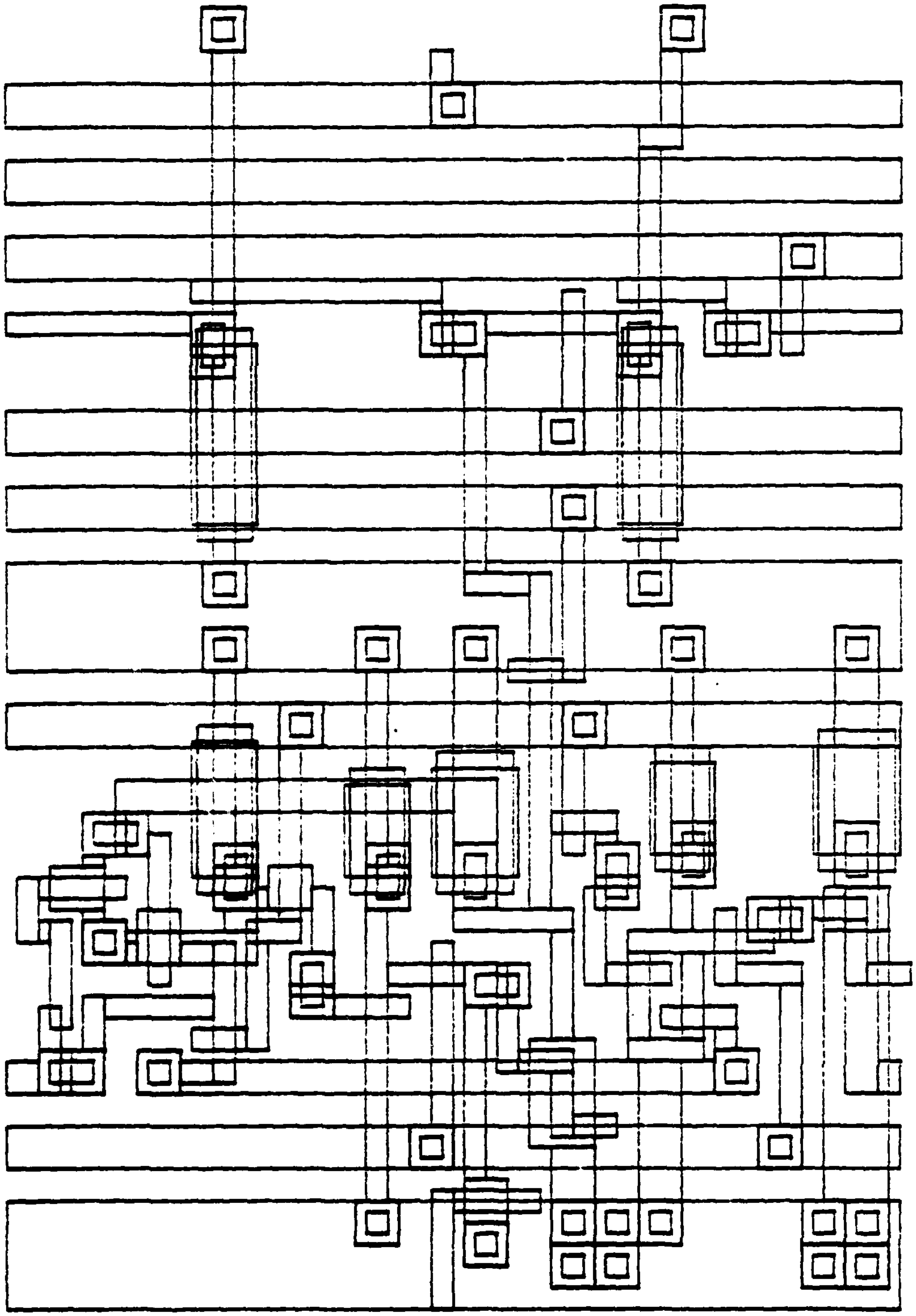
It appears that the future of NMOS is quite promising and the multiproject chip concept enables more people to use this technology. As feature sizes become smaller and more devices are packed on a chip, a structured design methodology becomes important to keep the design task manageable.

APPENDIX

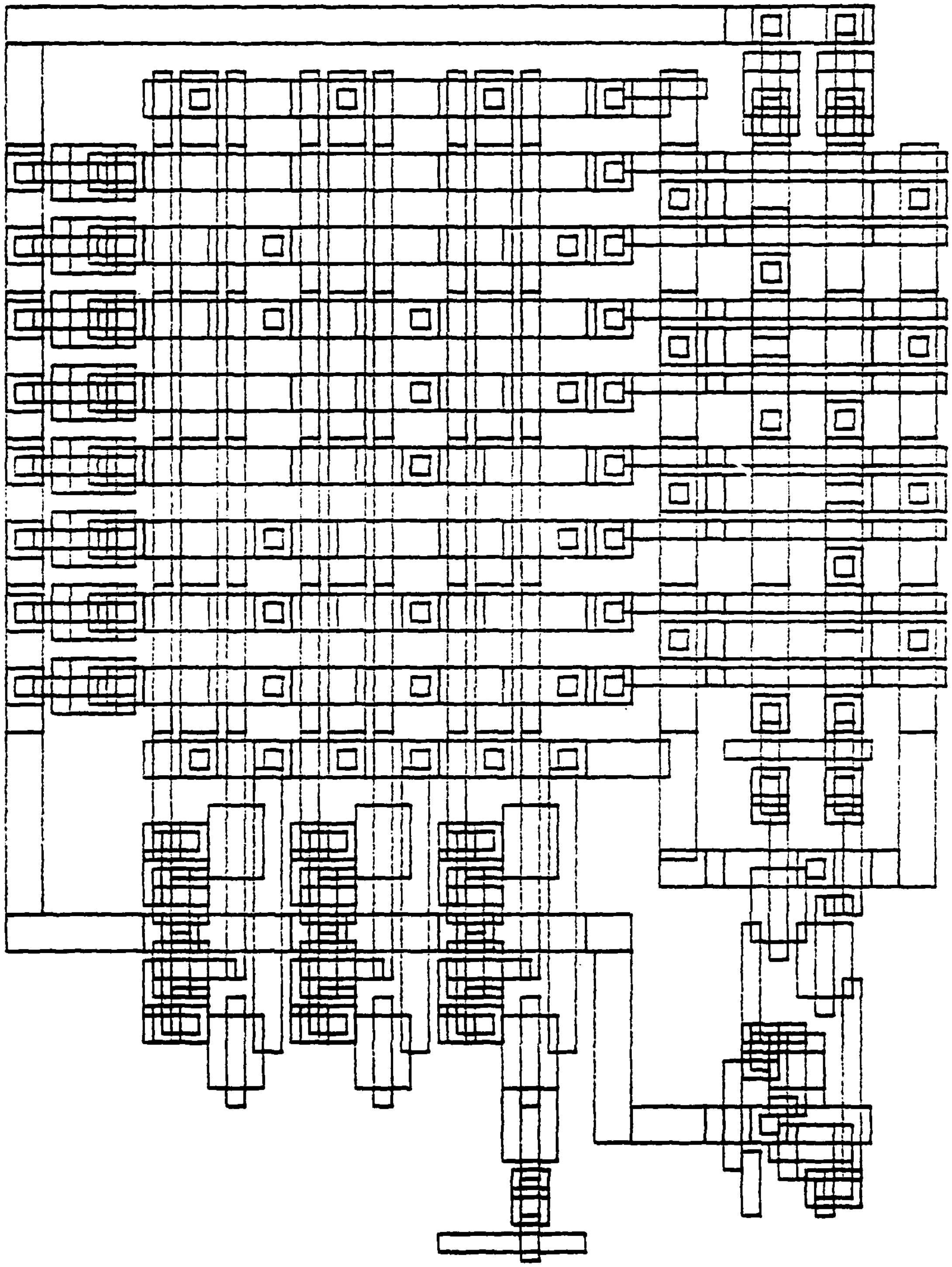
MASK LEVEL DESIGNS OF SELECTED NMOS CELLS



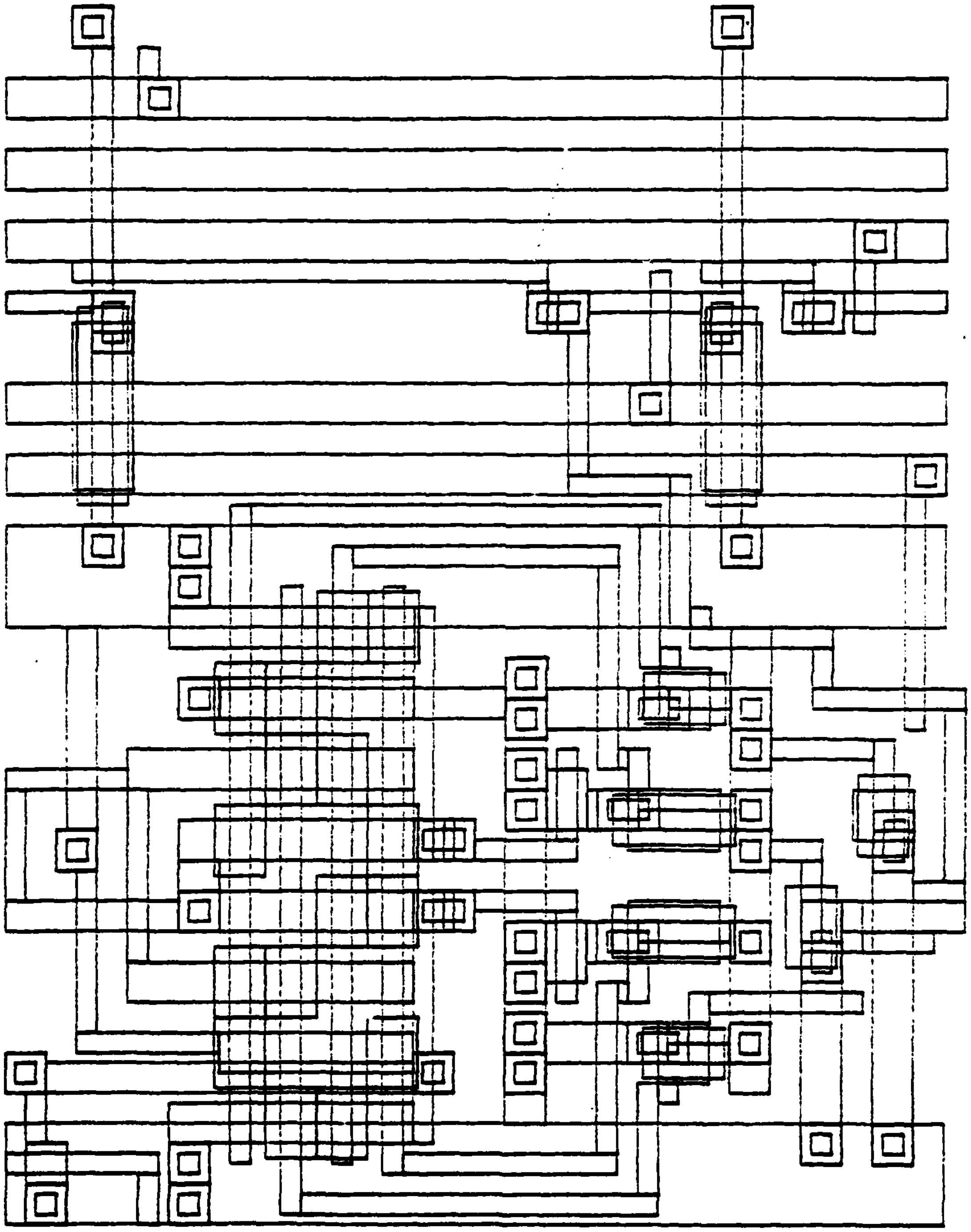
HOLD cell



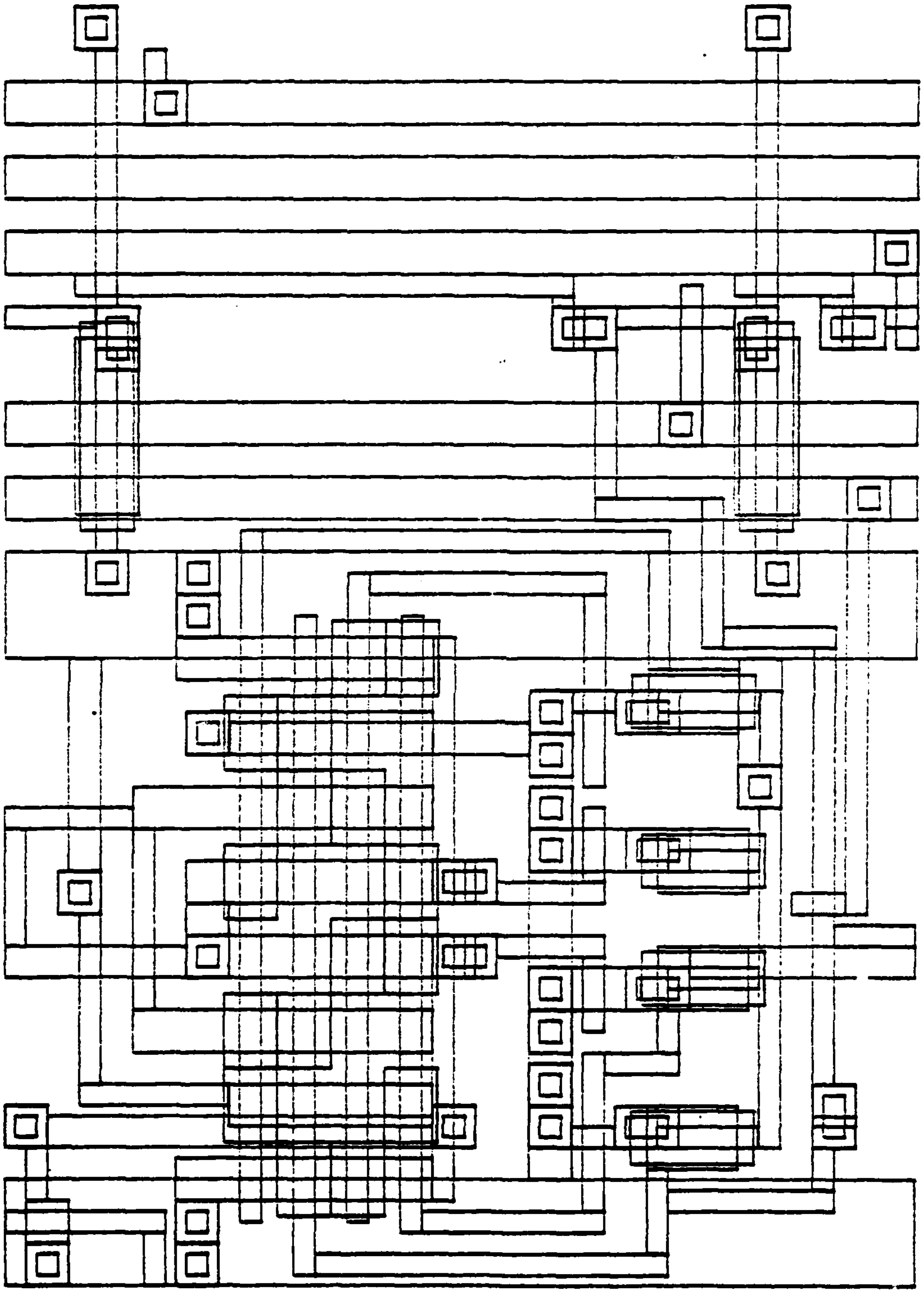
COUNT cell



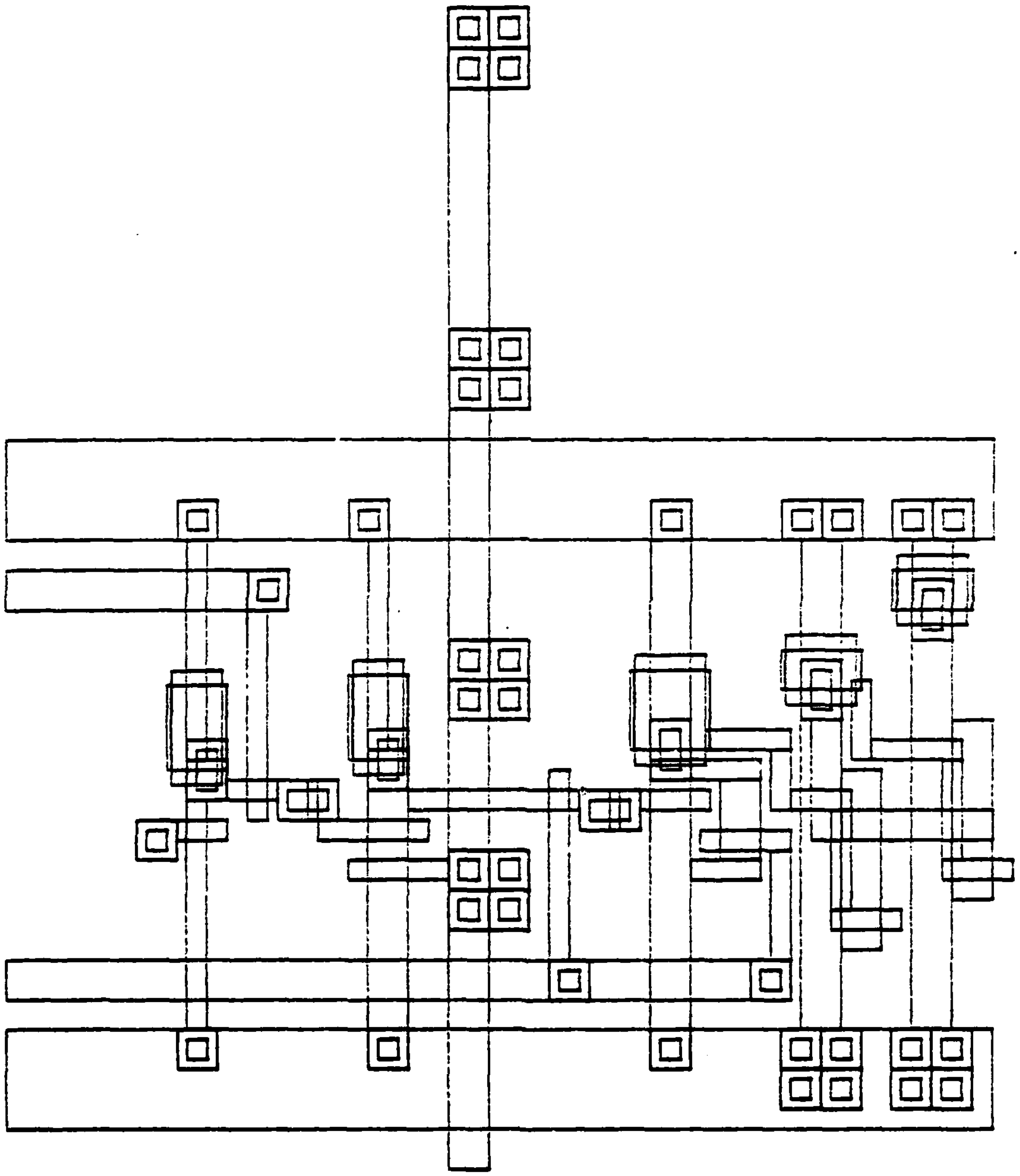
Serial one bit adder



High speed asynchronous counter cell



Low power asynchronous counter cell



HPUL cell

REFERENCES

1. C. Mead and L. Conway, Introduction to VLSI Systems, Addison-Wesley Publishing Company, Philippines, 1980.

2. L. Conway, A. Bell, and M. Newell, "MPC79: The Large-Scale Demonstration of a New Way to Create Systems in Silicon," Lambda, Second Quarter 1980, pp10-19.

3. B. Cooper, "Correlators with Two-Bit Quantization," Aust. J. Phys., v. 23, 1970, pp521-527.

4. Texas Instruments Incorporated, The TTL Data Book for Design Engineers, Texas Instruments Incorporated, Second Edition, 1976.

5. L. Nagel and D. Pederson, "Simulation Program with Integrated Circuit Emphasis (SPICE)," 16th Midwest Symposium on Circuit Theory, Waterloo, Ontario, April 12, 1973.